



INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

Monitorização e Detecção Automática de Anomalias em Redes IP

Michaël Lima Paiva de Castro

Dissertação para obtenção do Grau de Mestre em
Engenharia Electrotécnica e de Computadores

Júri

Presidente: Prof. Doutor Mário Serafim dos Santos Nunes

Orientador: Prof. Doutor Fernando Henrique Corte Real Mira da Silva

Co-Orientador: Prof.^a Doutora Teresa Maria Sá Ferreira Vazão Vasques

Vogais: Prof. Doutor Renato Jorge Caldeira Nunes

Setembro de 2008

Agradecimentos

Ao longo deste trabalho, várias foram as pessoas que contribuíram, directa ou indirectamente, com a sua ajuda e opinião. Quero desde já deixar a minha palavra de agradecimento.

Ao Prof. Fernando Mira da Silva, orientador, pela confiança e incentivo demonstrados e pelas suas ideias e sugestões no desenrolar do trabalho, permitindo a elaboração de novos objectivos e funcionalidades ao longo de todo este ano.

Ao Jorge Matias, administrador de sistemas do CIIST, pela disponibilidade e simpatia demonstradas na clarificação de quaisquer dúvidas sobre a rede do IST ou sobre seu funcionamento.

Aos restantes trabalhadores e bolseiros do CIIST, pela boa disposição e humor sempre presentes, providenciando momentos de descontração e bem-estar, ajudando assim a manter a motivação e moral.

Aos meus pais, pela dedicação e paciência demonstradas e pelo incentivo e apoio, sempre presentes, mesmo nos momentos menos positivos em que o pessimismo ameaçava denegrir o bom desenrolar do trabalho.

Aos meus colegas e amigos, pela compreensão e apoio na frequente falta de disponibilidade devido a este período de trabalho intenso.

Resumo

O conhecimento da topologia das redes *Ethernet* consiste numa das bases fundamentais para uma gestão adequada e uma correcta identificação de problemas. Sendo este conhecimento essencial, é necessário mantê-lo constantemente disponível e actualizado. No entanto, com o tamanho e complexidade das redes actuais, torna-se difícil e demorada a sua procura manual. Por outro lado, diversos são os riscos e perigos que ameaçam denegrir o bom funcionamento de uma rede. Embora o tráfego hostil seja frequentemente diferente do tráfego normal, torna-se difícil a sua tradução num conjunto de regras explícitas, devido ao tráfego ser fortemente irregular, variando assim os padrões da rede e os efeitos das anomalias.

Neste projecto, propõe-se uma nova ferramenta visando auxiliar no controlo e gestão de redes. Numa primeira fase, apresenta-se um algoritmo de pesquisa permitindo realizar uma descoberta automática da topologia de nível-2 da rede. Esta funcionalidade baseia-se em informação recolhida por SNMP, fornecida por MIBs de referência, sendo assim suportada pela grande maioria dos equipamentos. Seguidamente, descreve-se um método de aprendizagem não supervisionada que possibilita, de forma autónoma, a detecção de situações de funcionamento anómalo. As irregularidades são identificadas por um modelo de tráfego composto por uma mistura de gaussianas, responsável por caracterizar o funcionamento normal do sistema.

Uma das principais mais valias deste algoritmo é a sua adaptabilidade às condições da rede, tendo produzido bons resultados mediante variados cenários de funcionamento. Na aplicação à rede do IST, todas as ligações entre equipamentos foram devidamente identificadas, assim como diversas anomalias detectadas, para diversas configurações e níveis de carga na rede. Providencia-se assim uma reacção mais rápida às situações problemáticas na rede, sendo o controlo mais eficaz, tanto ao nível da gestão operacional, como ao nível da segurança dos sistemas.

Palavras-chave: Gestão de Redes, Monitorização de Redes, Descoberta da Topologia, Detecção de Anomalias, Segurança da Informação, SNMP.

Abstract

The knowledge of network topology is fundamental for a proper network management and problem identification. It's important to keep the information constantly updated in order for it to remain reliable and quickly available. However, the size and complexity of nowadays networks make the manual operation difficult and time-consuming, aiming the focus towards complex automated solutions. Furthermore, several are the risks and dangers that threaten the proper operation of a network. Even if hostile traffic is often different from benign traffic, it is hard to translate this difference in a set of explicit rules. This is mainly due to the highly irregular nature of the traffic, which constantly transforms network patterns and anomaly effects.

This project presents a new tool designed to assist in monitoring and network management. First, an algorithm is developed in order to allow automatic level-2 topology discovery. This feature is based on information collected by SNMP, provided by reference MIBs with the purpose of being supported by the vast majority of the equipments. The second part of this project describes a method of unsupervised learning which allows the detection of abnormal situations in the network. These irregularities are identified by a traffic model composed by a Gaussian mixture, responsible for representing the normal operation of the system.

One of the main advantages of this algorithm is its adaptability to a variety of network conditions, producing valuable results through different scenarios. The application at the IST network revealed all the network connections between equipments, as well as various network anomalies. These results prove that this tool enables a faster operation management, increasing system's efficiency and security.

Keywords: Network Management, Network Monitoring, Topology Discovery, Anomaly Detection, Information Security, SNMP.

Índice

Agradecimentos	i
Resumo.....	iii
Abstract	iv
Índice	v
Lista de Figuras	viii
Lista de Tabelas.....	x
Lista de Acrónimos	xi
1. Introdução.....	1
2. Tecnologias e Ferramentas de Gestão de Redes.....	3
2.1. Gestão de Redes	3
2.2. SNMP	4
2.2.1. Descrição	4
2.2.2. Modelo da Informação	5
2.2.3. Versões	7
2.2.4. <i>Polling</i>	7
2.2.5. Interrupções	8
2.3. Ferramentas de Monitorização e Gestão.....	9
2.3.1. Introdução	9
2.3.2. Nagios.....	9
2.3.3. HP-OpenView	10
2.3.4. Ferramentas Dedicadas.....	12
3. Descoberta da Topologia de Rede	13
3.1. Introdução	13
3.2. Estado da Arte	15
3.2.1. Trabalhos Relacionados	15
3.2.2. Metodologias Seleccionadas	16
3.3. Algoritmo Realizado	17
3.3.1. Fundamento.....	17
3.3.2. Ligações Obtidas a Partir de AFTs Completas.....	19
3.3.3. Pesquisa de Caminhos.....	21

3.3.4.	Ordenação de Caminhos	23
3.3.5.	Dedução de Ligações Directas	26
3.3.6.	Hierarquia do Equipamento	27
4.	Monitorização da Rede	31
4.1.	Recolha de Informação	31
4.2.	Armazenamento dos Dados	32
4.2.1.	Introdução	32
4.2.2.	Modelo de Dados	32
4.2.3.	Acesso à Informação	33
4.3.	Possibilidades de Gestão	34
4.3.1.	Localização de Equipamentos	34
4.3.2.	Análise de Tráfego	36
5.	Detecção de Anomalias	39
5.1.	Introdução	39
5.2.	Estado da Arte	40
5.2.1.	Metodologias Desenvolvidas	40
5.2.2.	Fundamento Utilizado	41
5.3.	Modelo de Tráfego	42
5.3.1.	Parâmetros de Entrada	42
5.3.2.	Algoritmo Expectation-Maximization	43
5.3.3.	<i>Minimum Message Length</i>	44
5.3.4.	Identificação de Anomalias	47
5.4.	Aplicações à Gestão de Redes	48
6.	Apresentação dos Resultados	51
6.1.	Interface do Utilizador	51
6.1.1.	Página Web	51
6.1.2.	Representação da Topologia	52
6.1.3.	Apresentação das Anomalias	53
6.2.	Resultados dos Testes	54
6.2.1.	Descoberta da Topologia	54
6.2.2.	Localização de Equipamentos na Rede	58
6.2.3.	Detecção de Anomalias	60
6.2.4.	Identificação de Anomalias	64
7.	Conclusões	69
7.1.	Observações Finais	69
7.2.	Aplicações Futuras	70

A. Manual técnico	73
B. Manual utilizador.....	77
Referências	83

Lista de Figuras

Figura 2.1 – Detalhe das ligações da <i>Management Information Base</i> , nomeadamente em direcção à MIB-2.	6
Figura 2.2 - Percurso da raiz até à <i>Management Information Base</i> , na estrutura em árvore dos objectos alcançáveis por SNMP.	6
Figura 3.1 - Exemplo de uma rede contendo vários <i>switches</i>	20
Figura 3.2 - Descoberta de ligações a partir de AFTs completas.	21
Figura 3.3 – Procura de caminhos a partir de AFTs potencialmente incompletas.	23
Figura 3.4 – Ordenamento de caminhos com base em ligações garantidas.	24
Figura 3.5 – Ordenamento de caminhos com base em AFTs.	25
Figura 3.6 – Exemplo de uma cascata de <i>switches</i>	26
Figura 3.7 – Dedução das ligações entre <i>switches</i> a partir dos caminhos.	27
Figura 3.8 – Dedução da hierarquia de rede.	29
Figura 3.9 – Exemplo de uma rede em estrela com um <i>router</i> ao centro ligado à Internet.	30
Figura 4.1 – <i>Query</i> à base de dados para obter o porto de saída de um <i>switch</i>	34
Figura 4.2 – Localização de um equipamento na rede pelo endereço MAC.	35
Figura 6.1 – Consola de gestão – janela relativa ao estado do <i>polling</i> à rede.	52
Figura 6.2 – Evolução do tráfego entre 09/07/08 18:00 e 13/07/08 18:00, na interface 2010 do <i>router</i> central do IST.	53
Figura 6.3 – Imagem da topologia da rede, em Janeiro de 2008, representando os equipamentos pelos respectivos endereços IP.	54
Figura 6.4 – Imagem da topologia da rede, em Janeiro de 2008, representando os equipamentos pelos respectivos nomes configurados.	55
Figura 6.5 – Imagem da nova topologia da rede, em Junho de 2008, representando os equipamentos pelos respectivos endereços IP.	56
Figura 6.6 – Imagem da nova topologia da rede, em Junho de 2008, representando os equipamentos pelos respectivos nomes configurados.	57
Figura 6.7 – Localização de um equipamento na rede, a partir do seu endereço ip.	58
Figura 6.8 – Localização de um equipamento na rede, a partir do seu endereço físico.	58
Figura 6.9 – Imagem dinâmica da topologia da rede, centrada no <i>switch</i> “192.168.249.222”, contemplando as ligações até ao core da rede (“192.168.249.1”).	59
Figura 6.10 – Localização de um equipamento na rede, ligado a um <i>switch</i> intermédio.	59
Figura 6.11 – Localização de um equipamento, ligado na periferia da rede.	60
Figura 6.12 – Detecção de anomalias baseada no modelo 3.	64
Figura 6.13 – Informação sobre os parâmetros do modelo de tráfego.	65
Figura 6.14 – Tráfego no porto 1001 do <i>router</i> central, evidenciando a anomalia de dia 15 de Julho.	66
Figura 6.15 – Tráfego no porto 2008 do <i>router</i> central, destacando a anomalia de dia 19 de Julho. ...	66
Figura A.1 – Ficheiro de configuração da ferramenta criada.	74

Figura A.2 – Ficheiro de configuração da interface Web.	75
Figura B.1 – Descoberta da rede, na interface Web.	77
Figura B.2 – Anomalias detectadas na rede.	78
Figura B.3 – Informação sobre a rede, na interface Web.	79
Figura B.4 – Informação sobre as VLANs, na interface Web.	80
Figura B.5 – Topologia de rede, na interface Web.	81

Lista de Tabelas

Tabela 6.1 – Tráfego considerado normal pelo modelo 1.....	61
Tabela 6.2 – Tráfego considerado anómalo pelo modelo 1.....	61
Tabela 6.3 – Tráfego considerado normal pelo modelo 2.....	62
Tabela 6.4 – Tráfego considerado anómalo pelo modelo 2.....	63
Tabela 6.5 – Detecção de anomalias a partir do modelo 3.....	63

Lista de Acrónimos

AFT	<i>Address Forwarding Table</i>
AP	<i>Access Point</i>
ARP	<i>Address Resolution Protocol</i>
DOS	<i>Denial of Service</i>
EM	<i>Expectation-Maximization</i>
ICMP	<i>Internet Control Message Protocol</i>
IDS	<i>Intrusion Detection Systems</i>
IPS	<i>Intrusion Prevention System</i>
IETF	<i>The Internet Engineering Task Force</i>
IP	<i>Internet Protocol</i>
LAN	<i>Local Area Network</i>
MAC	<i>Media Access Control</i>
MIB	<i>Management Information Base</i>
MML	<i>Minimum Message Length</i>
N/A	<i>Non Available</i>
NMS	<i>Network Management System</i>
NNM	<i>Network Node Manager</i>
OID	<i>Object Identifier</i>
OSI	<i>Open Systems Interconnection</i>
PC	<i>Personal Computer</i>
RFC	<i>Request for Comments</i>
SMI	<i>Structure of Management Information</i>
SNMP	<i>Simple Network Management Protocol</i>
SQL	<i>Structured Query Language</i>
SSH	<i>Secure Shell</i>
SSL	<i>Secure Sockets Layer</i>
STP	<i>Spanning Tree Protocol</i>
TCP	<i>Transmission Control Protocol</i>
UDP	<i>User Datagram Protocol</i>
UML	<i>Unified Modeling Language</i>
VLAN	<i>Virtual Local Area Network</i>
XML	<i>eXtensible Markup Language</i>

1. Introdução

Hoje em dia, poucas são as pessoas que não utilizam telemóveis e numerosas aquelas que optam pelo uso diário de computadores. A sociedade actual tornou-se dependente das novas tecnologias, tanto na área das telecomunicações como da informática, situação que tem vindo a massificar-se cada vez mais rapidamente. Com o crescente número de aplicações e serviços, torna-se necessário proceder à troca e partilha de informação e, deste modo, dar cada vez mais ênfase à computação em rede. Aparecem, por consequência, novos requisitos de desempenho, qualidade de serviço e segurança, no intuito de proporcionar mais e melhores funcionalidades, não deixando de ter em atenção os novos riscos que possam surgir. O resultado consiste num aumento significativo do tamanho e complexidade das redes, originando um incremento nos recursos e trabalho de administração necessários.

Para garantir um funcionamento constante dos equipamentos, as redes necessitam de ser cuidadosamente configuradas, monitorizadas e geridas. É imprescindível não só controlar o estado do tráfego e dos serviços existentes, mas também intervir na rede quando necessário, de forma a manter as funcionalidades previstas, mesmo quando surgem falhas ou anomalias inesperadas. Sem uma prevenção adequada e sem acções correctivas eficazes, o desempenho de uma rede encontra-se em risco, pois qualquer irregularidade pode facilmente degradar ou mesmo suspender a normal operacionalidade dos sistemas.

Uma rede IP (*Internet Protocol*) apresenta, frequentemente, problemas originados por modificações e alterações da sua topologia e infra-estrutura. A instalação ou migração de novos equipamentos, assim como mudanças nas configurações existentes, podem criar instabilidade na rede e provocar falhas em inúmeros serviços. Esta situação é agravada pela heterogeneidade e complexidade das redes que dificultam ainda mais as tarefas de gestão em tempo útil. Uma visão global da rede, contendo informação sobre as ligações, configurações e funcionamento de cada equipamento, torna-se assim vital para uma gestão correcta e eficiente.

Sendo a rede do IST uma rede tecnologicamente heterogénea, complexa e em constante mutação, pretende-se com este trabalho desenvolver uma nova ferramenta, centralizada e permanentemente actualizada, que permita uma gestão da rede mais rápida e eficaz, possibilitando uma melhor qualidade de serviço prestado e um aumento na segurança global.

A rede do IST Alameda, rede IP de média dimensão em constante evolução, é, no geral, gerida de forma descentralizada, sendo delegada a vários responsáveis. Deste modo, facilmente se perde o controlo de toda a rede, em particular das extremidades, já que nem sempre os gestores locais informam os restantes das alterações efectuadas na sub-rede que lhes pertence. Um sistema de

informação central, contendo informação actualizada sobre a rede, a sua topologia e a sua actividade, é crucial para uma rápida detecção, localização e resolução de qualquer situação problemática.

Pretende dar-se uma ênfase especial à descoberta da topologia de nível 2, para permitir a detecção automática das suas alterações, funcionalidade actualmente inexistente no IST. Numa segunda fase, procura-se aumentar o conhecimento sobre o estado de funcionamento da rede, nomeadamente a partir dos valores de tráfego existentes, criando-se assim uma base de dados flexível e utilizável em operações de análise e tratamento de dados, visando a detecção de irregularidades.

Diversas são as ameaças ao bom funcionamento de uma rede. Esta pode ser afectada não só por problemas operacionais, tais como erros de configuração, avarias ou falhas nas inter-ligações, mas também por ataques ou violações de segurança, originadas, nomeadamente, por aplicações maliciosas. Devido à multiplicidade de riscos, torna-se complexo catalogar todas as situações irregulares, dificultando assim a sua detecção. Propõe-se, neste projecto, abordar o tema de outra perspectiva, caracterizando o funcionamento normal do sistema, detectando à posteriori situações indiciando potenciais irregularidades.

O objectivo desta componente consiste no desenvolvimento de um mecanismo de aprendizagem automática, que permita a criação de um modelo contemplando as diferentes variações de tráfego na rede. Dessa forma, visam detectar-se situações de funcionamento anómalo, mesmo de origem desconhecida, resultantes de um afastamento relativamente aos valores previstos pelo modelo alcançado. A análise realiza-se assim ao nível da rede, procurando prevenir fortes variações que possam provocar uma diminuição ou mesmo uma interrupção na qualidade de serviço.

Este trabalho foi desenvolvido tendo como referencia a rede do IST Alameda, dado ser este o contexto de desenvolvimento e de aplicação. No entanto, as técnicas e metodologias utilizadas podem facilmente ser aplicadas a outras redes com características similares, sendo a portabilidade uma das preocupações principais deste projecto.

O restante deste relatório está estruturado como se segue. No capítulo 2, inicia-se o estudo fazendo uma pesquisa sobre as tecnologias e ferramentas existentes, verificando quais as mais adequadas para a resolução do problema em estudo. No capítulo 3, implementa-se a descoberta automática da topologia de nível 2 da rede. O capítulo 4 é dedicado à monitorização da rede e às possibilidades de gestão oferecidas, não deixando de frisar o modo de armazenamento de toda esta informação. Este conjunto de dados é então utilizado no capítulo 5, onde se descrevem as metodologias desenvolvidas para permitir a detecção de anomalias na rede. No capítulo 6, apresentam-se os resultados obtidos, assim como as possibilidades de acesso aos mesmos por parte do utilizador. Por último, o capítulo 7 conclui o relatório, apresentando as observações finais deste projecto. Em anexo encontram-se igualmente os manuais da aplicação desenvolvida, para qualquer detalhe adicional sobre as funcionalidades disponibilizadas.

2. Tecnologias e Ferramentas de Gestão de Redes

2.1. Gestão de Redes

As redes foram inicialmente criadas com o objectivo de permitir a troca e partilha de recursos numa dinâmica de inter-operação. No presente, o protocolo mais utilizado globalmente na transmissão de dados é o TCP/IP (*Transmission Control Protocol / Internet Protocol*) [SoKa91]. O TCP/IP corresponde, na realidade, a um dos protocolos mais importantes nas arquitecturas de rede devido à sua adopção e aceitação pela Internet, tema de [KuRo04] e [Tane02].

Na fase inicial do TCP/IP, não existiam mecanismos de gestão. Nessa altura, as redes eram pouco complexas, com um número reduzido de equipamentos e com funcionalidades e acesso limitados a um conjunto restrito de utilizadores. Era então possível manter o funcionamento das redes, efectuando apenas uma gestão individualizada e dedicada aos equipamentos. Uma das poucas ferramentas existentes, apesar de limitada, era o *Internet Control Message Protocol* (ICMP) [Post81] que tornava possível monitorizar a acessibilidade e o atraso na rede graças, por exemplo, ao utilitário PING.

Com a expansão exponencial que se fez sentir a partir dos anos 80, as redes actuais tornaram-se muito diferentes das existentes na sua génese. Ganharam uma nova dimensão e novas propriedades. Uma rede típica actual caracteriza-se pela diversidade, complexidade e capacidade de interligação, independentemente da tecnologia. Este grande desenvolvimento permitiu uma democratização das redes, com a consequente proliferação de equipamentos e serviços a baixo custo. Os utilizadores são cada vez mais numerosos e, com o melhoramento constante dos serviços, exigem um crescendo de fiabilidade e desempenho.

Com este crescimento, a gestão individualizada deixa de ser possível pois os problemas passam a estar ligados a vastas redes com equipamentos dotados de novas e mais funcionalidades. Sem uma visão global da rede, os problemas tornam-se dificilmente resolúveis. Por outro lado, os custos a que este método obrigaria, dada a dimensão e complexidade da rede, tornam este modelo de gestão economicamente inviável. O ICMP passa a ser insuficiente para controlar e analisar as redes tendo em conta os novos requisitos de qualidade de serviço e as novas responsabilidades. Sendo o mercado responsável pela competitividade e subsistência das empresas, é necessário criarem-se novas ferramentas de gestão, mais completas e eficazes, para se poder fazer face à concorrência. A importância e a responsabilidade do bom funcionamento de uma rede passam a ser prioritárias.

Assim nasce o verdadeiro conceito de gestão de redes que consiste no uso de diversas ferramentas, técnicas e sistemas que permitam gerir e controlar equipamentos e serviços, garantindo assim uma melhor operacionalidade e qualidade. Cabe aos administradores de redes modelar, rentabilizar e otimizar o novo leque de possibilidades que a gestão de redes lhes traz [Vaza04].

Desenvolve-se então o protocolo *Simple Network Management Protocol* (SNMP), [CFSD90], que vem colmatar a falta de soluções na recolha e organização de informação da rede. Proporcionam-se assim várias possibilidades e soluções de gestão de redes, permitindo a criação de variadas ferramentas, entre as quais se encontram as mais utilizadas actualmente. Sendo, presentemente, o protocolo de referência na gestão de redes IP, este é suportado e implementado pela grande maioria dos equipamentos, o que permite efectuar uma gestão remota e de alto nível, baseando-se apenas nesta tecnologia.

2.2. SNMP

2.2.1. Descrição

O SNMP corresponde a um modelo de troca de informações pela rede que permite aos administradores monitorizar e eventualmente efectuar operações de gestão nas máquinas que o suportam [MaSc05]. Este protocolo foi desenvolvido pelo IETF (*The Internet Engineering Task Force*) no RFC (*Request for Comments*) 1157 [CFSD90].

A gestão SNMP baseia-se numa arquitectura cliente-servidor na qual o cliente corresponde ao gestor e o servidor ao agente. Neste modelo, o gestor, *Network Management System* (NMS), é responsável pela obtenção da informação dos agentes da rede e na apresentação dos resultados numa interface com o utilizador. A informação obtida possibilita a aquisição de conhecimentos sobre o estado do equipamento e a sua actividade, permitindo, deste modo, adquirir dados estatísticos da rede ou mesmo detectar a ocorrência de alguma falha ou anomalia. Em seguida, torna-se possível enviar comandos por SNMP, de modo a gerar sinalização ou a actuar directamente no equipamento, visando a resolução dos problemas detectados. Por outro lado, o agente é um *software* que corre nos equipamentos com o objectivo de manter uma base de dados dinâmica contendo informações sobre o estado e desempenho dos mesmos.

O gestor acede à informação por meio de uma *Management Information Base* (MIB) [RoMc90b]. Esta informação pode ser requisitada periodicamente pelo gestor - *polling*¹ - ou enviada directamente pelo agente, na ocorrência de um evento pré-programado para tal – interrupção (*trap*).

¹ Neste contexto, *polling* pode ser traduzido por “consulta” em Português. No entanto, não havendo uma tradução normalizada para esta expressão, preferiu-se aqui a utilização do termo anglo-saxónico original.

2.2.2. Modelo da Informação

Para uniformizar a informação partilhada e possibilitar a sua troca entre equipamentos de vários fabricantes, o SNMP usa a sintaxe definida pela *Structure of Management Information* (SMI) [RoMc90a]. Nesse contexto, criam-se as MIBs que são conjuntos de definições em SMI dos objectos a que o agente tem acesso. Correspondem estes a variáveis físicas ou lógicas do equipamento que podem ser úteis em tarefas de gestão. Tomando o exemplo de um *router*, um objecto pode corresponder, para além de muitas possibilidades, ao número de interfaces que este possui ou ao tráfego recebido num dado porto, informação frequentemente útil para o administrador de sistemas. As MIBs associam um nome textual à uma variável do sistema gerido e explicam o seu significado.

Um agente pode implementar diversas MIBs. Existem MIBs padrão que todos suportam e MIBs proprietárias que são definidas e implementadas pelos diversos fabricantes mas apenas suportadas por estes. Estas últimas são no geral mais específicas e detalhadas por serem adaptadas ao *hardware* gerido, mas não são utilizáveis numa rede heterogénea para a qual se pretende efectuar uma gestão centralizada de todos os equipamentos. É frequente ser utilizada a MIB-II, definida em [McRo91], que consiste numa MIB de referência porque é implementada, mesmo que nem sempre totalmente, por todos os equipamentos suportando SNMP. Esta contém objectos com as principais características pretendidas para monitorização e gestão, sendo regularmente suficiente para obter toda a informação desejada.

Um agente implementando uma MIB pode ser comparado a uma base de dados dinâmica, constantemente actualizada e cujo conteúdo corresponde aos objectos da MIB. O gestor apenas tem acesso à informação contida nas MIBs do agente e tem, por isso, de conhecer previamente a estrutura e organização das MIBs, antes de poder utilizar a informação recolhida.

As MIBs são estruturadas hierarquicamente em árvore fazendo-se corresponder as folhas aos objectos geridos e os restantes nós à organização e separação dos tipos de variáveis. Cada objecto tem associado um *Object Identifier* (OID), identificador que pode aparecer na forma textual ou numérica. A cada nível da árvore, desde a raiz até às folhas, para cada novo OID a numeração incrementa de 1. Se considerarmos o exemplo da MIB-II, tal como se pode observar na Figura 2.1, esta encontra-se localizada em 1.3.6.1.2.1 (OIDs na forma numérica). Esta nomenclatura significa que a MIB-II encontra-se no nó 1 da raiz, no nó 3 seguinte e nos respectivos nós restantes - Figura 2.2.

Para aceder e modificar os objectos das MIBs, o SNMP possui um conjunto de operações que podem ser efectuadas por comandos, *scripts* ou outras ferramentas. Destacam-se: o GET que permite obter um objecto, dado o seu OID, tornando assim possível efectuar monitorização, o SET que permite a escrita num objecto e deste modo controlar e gerir o equipamento em questão e o TRAP que

corresponde à informação enviada directamente pelo agente no caso do evento previamente configurado ser despoletado.

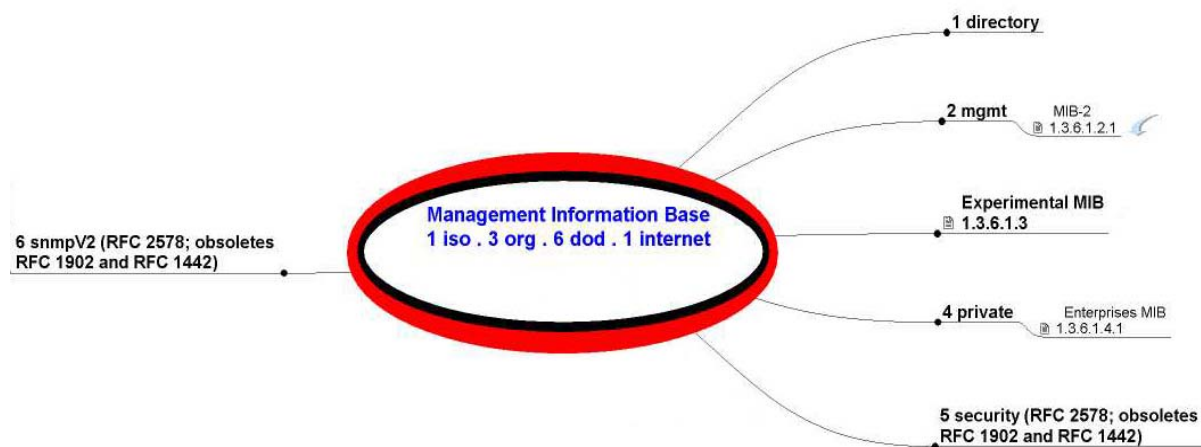


Figura 2.1 – Detalhe das ligações da *Management Information Base*, nomeadamente em direcção à MIB-2.

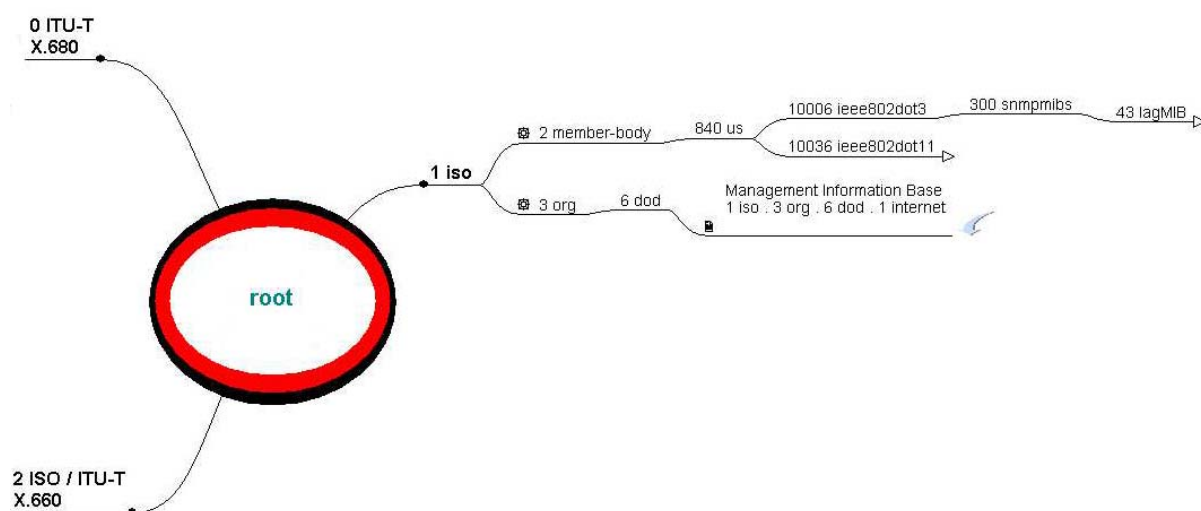


Figura 2.2 - Percurso da raiz até à *Management Information Base*, na estrutura em árvore dos objectos alcançáveis por SNMP.

Como já foi referido, a rede do IST é composta por diversos equipamentos de diferentes fabricantes e, como tal, não é possível utilizar objectos de MIBs proprietárias para obter informação em toda a rede. No entanto, todos os dados necessários para a execução deste projecto podem ser obtidos a partir da MIB-2, suportada por todos os equipamentos do IST. Por este motivo, foi esta a MIB escolhida para o restante desenrolar do projecto.

2.2.3. Versões

Existem várias versões de SNMP em razão dos constantes melhoramentos criados para fazer face a novas exigências, nomeadamente de segurança e funcionalidade.

A versão inicial do SNMP (SNMPv1) baseia o acesso aos dados por comunidades (*communities*), que podem ser equiparadas a palavras passe em texto. Tipicamente existem três comunidades: de leitura, de escrita e de interrupção. Não existe segurança real neste protocolo pelo facto de apenas ser necessário conhecer a comunidade para aceder à informação de gestão de um equipamento e esta ser transmitida em claro pela rede. Mesmo estando desactualizada para os requisitos de segurança mais exigentes, continua a ser a primeira versão suportada pelos fabricantes.

A versão 2 (SNMPv2) veio aumentar as possibilidades do SNMP estendendo o SMI para permitir novos tipos de dados e outros ramos na estrutura em árvore das MIBs. Esta versão vem também resolver algumas limitações do SNMPv1 tais como o suporte a redes complexas, com um número elevado de agentes e um grande volume de informação transferida. Por outro lado, existe um desenvolvimento paralelo focado na segurança mas sem tomar em conta os melhoramentos de desempenho, ficando este assim limitado como o SNMPv1.

Partindo do SNMPv2 com os melhoramentos de desempenho, o SNMPv3 veio resolver a questão de segurança. O SNMPv3 passa a ser responsável não só pela troca de mensagens SNMP, mas também pela autenticação, cifra, decifra e controlo de acesso aos objectos geridos. Torna-se, assim, finalmente possível autenticação forte e comunicações privadas entre o gestor e os agentes.

Contudo, a rede do IST Alameda dispõe de equipamentos que apenas suportam SNMPv1 ou SNMPv2 sem segurança. Por conseguinte, como o objectivo deste trabalho consiste em criar uma ferramenta que comunique com todo o equipamento de rede, vai ser escolhida a versão mais simples, o SNMPv1, que não deixa de preencher os requisitos necessários e corresponde à única possibilidade suportada por todas as máquinas do IST.

2.2.4. Polling

Para obter a informação necessária, o gestor pode enviar um pedido ao agente contendo o OID do objecto desejado e esperar pela resposta. A este processo dá-se o nome de *polling*. Desde modo, o gestor consegue ter acesso a todos os objectos implementados pelo agente, na altura que achar adequada. Todavia, o gestor necessita, na maioria das vezes, de ter acesso a uma informação actualizada, idealmente em tempo real, sobre o estado do equipamento. No caso de um *router* central de uma rede, se uma interface deixar de funcionar como previsto, o gestor tem interesse em ter conhecimento dessa falha o mais depressa possível para poder iniciar a sua resolução.

Uma das maneiras de resolver este problema é efectuar *polling* periodicamente na rede. Este método consiste em realizar operações de GETs dos objectos mais importantes para gestão, espaçados por um intervalo de tempo a definir. Este intervalo é escolhido consoante a importância do objecto e a carga na rede: um sistema vital ao funcionamento da rede necessita de um intervalo de *polling* muito reduzido enquanto que a informação estatística, por exemplo, pode ser efectuada apenas uma vez por dia.

A comunicação entre gestor e agente passa pela rede, logo, não pode gerar tráfego que impeça o bom funcionamento da mesma. Um problema a ter em conta corresponde à situação de existirem vários elementos a serem monitorizados com um dado intervalo. Deve-se assegurar, especialmente numa rede com carga elevada, que o último objecto é obtido antes de se recomeçar o processo, ou seja, antes de se efectuar um novo *polling* da sequência. Caso contrário, falhas de coerência e de operacionalidade podem ocorrer no NMS.

2.2.5. Interrupções

Uma alternativa ao *polling* consiste em configurar o agente para ser ele a enviar directamente a informação ao gestor sem que este último a requisite. Os dados são enviados assincronamente e permitem deste modo alertar o gestor da ocorrência de eventos pré-definidos, em tempo real, razão pela qual se designam estes envios por interrupções. Este método possibilita uma redução de carga na rede e um aumento da velocidade de resposta, uma vez que uma interrupção é enviada mal seja despoletada, mas apenas aquando dessa ocorrência. No entanto, este método necessita de uma configuração prévia em cada equipamento em que estiver activo. Tanto esta configuração, como aquela necessária para qualquer alteração numa interrupção já existente, são significativamente mais demoradas e trabalhosas que o seu equivalente por *polling*, em que toda a configuração é feita uma só vez e do lado do gestor.

É aconselhado o uso de interrupções na monitorização da informação mais importante e vital da rede porque, caso contrário, esta necessitaria de um *polling* muito frequente para ser eficaz. Na rede restante, o uso moderado de *polling* é normalmente suficiente, proporcionando assim uma maior facilidade de controlo e configuração. Em ambos os casos, depois de recebido o objecto e processada a informação correspondente, é possível efectuarem-se operações de gestão, actuando em conformidade nos equipamentos por meio de comandos SET. Pode ser implementado um sistema que desactive um porto de um dado *switch*, por exemplo, se nele se verificar algum tráfego anormal, evitando assim uma degradação do funcionamento global.

2.3. Ferramentas de Monitorização e Gestão

2.3.1. Introdução

Existem várias ferramentas que utilizam o SNMP para efectuarem operações de monitorização e gestão. A maneira mais simples de aceder directamente aos objectos das MIBs consiste em utilizar o pacote Net-SNMP [Nets06], ou equivalente, que permite um acesso directo às primitivas fundamentais do protocolo SNMP, a partir de funções como *snmpget* ou *snmpset*. No entanto este método é muito ineficiente porque obriga o envio de um comando, com o OID do objecto desejado, para cada informação procurada. Como alternativa, é possível realizar a leitura de vários objectos a partir de um só comando: o *snmpwalk* (também disponível no pacote Net-SNMP). Este percorre a MIB em busca de objectos que verifiquem o nome ou OID introduzido retornando um ou mais resultados, se existentes. É igualmente possível introduzir nomes ou OID incompletos, uma vez que o comando procura todos os objectos que verifiquem esse fragmento. Todavia, esta metodologia apenas torna possível uma gestão pontual e individualizada da rede, por ser morosa e essencialmente manual.

Para ajudar na resolução deste problema, existem diversas ferramentas e métodos que permitem gerir e monitorizar a rede de uma forma mais eficiente e automatizada. Nas duas secções seguintes, abordam-se as duas ferramentas frequentemente adoptadas: o Nagios e o HP-OpenView. Em seguida, estuda-se como alternativa a criação de uma nova ferramenta, dedicada e adaptada às necessidades do IST.

2.3.2. Nagios

O Nagios é um programa de *software* aberto (*open-source*) [Gals06] que executa monitorização de serviços e não de rede, embora possa ser usada neste contexto [MaLo]. O Nagios serve-se de comandos e protocolos para testar o estado de diversos serviços tais como HTTP, FTP, SSH, PING, POP3 ou SMPT, sem necessitar de SNMP. A informação acedida, mesmo que limitada, torna possível obter uma visão geral dos serviços disponíveis da rede, assim como o seu desempenho, sem necessitar de configurar e instalar novo *software* nas máquinas alvo.

Esta ferramenta é dificilmente configurável, sendo necessário introduzir manualmente os equipamentos e serviços a monitorizar num formato nem sempre intuitivo, e está limitada às tarefas de escuta da rede, sendo a gestão apenas suportada por interacção com outras ferramentas. Não deixa no entanto de ser um *software*, depois de correctamente configurado, facilmente utilizável e acessível, graças a sua interface *Web* e à sua informação estatística apresentada de uma maneira visualmente atraente.

Já se encontra configurada e em funcionamento uma versão do Nagios na rede do IST, sendo utilizada para obter informação sobre o estado dos serviços e gerar SMS na ocorrência de alarmes pré-programados.

Em todo o caso, embora o Nagios permita a eventual detecção de problemas na rede por interrupção de serviços, o seu principal objectivo não é a monitorização do equipamento, não sendo consequentemente o mais adequado para o efeito. Não providenciando qualquer tipo de informação sobre topologia ou tráfego, são necessárias outras ferramentas para se realizarem todos os objectivos deste trabalho, as quais irão ser estudadas em seguida.

2.3.3. HP-OpenView

Esta ferramenta corresponde a um dos mais completos e complexos NMS existentes actualmente. Permite a gestão de redes de grande dimensão e complexidade, tais como as redes dos *Internet Service Providers* (ISP) ou outras ao nível nacionais. No entanto, não deixa de ser utilizável com sucesso em redes de média dimensão como, por exemplo, a do IST.

O HP-OpenView *Network Node Manager* (NNM) [HPOV] permite funcionalidades como descoberta, monitorização e gestão da rede, depois de configurado correctamente. A configuração para obter o funcionamento e dados desejados é complexa [HPOV03a], mas a interface do utilizador oferece os resultados de forma simples e clara. A informação recolhida da rede, assim como os históricos e os ficheiros de recuperação, são guardados numa base de dados proprietária com poucas possibilidades de exploração [HPOV03b]. Porém, os dados são exportáveis para as bases de dados comerciais Oracle e Microsoft SQL Server, podendo utilizar-se a informação resultante a partir de *scripts* ou outras ferramentas externas e assim permitir novas funcionalidades.

A descoberta da rede é realizada através de *polling*, utilizando os protocolos TCP/IP, ICMP, UDP (*User Datagram Protocol*), SNMP e as tabelas ARP (*Address Resolution Protocol*), que contêm informação sobre a tradução de endereços entre nível-2 – endereços MAC (*Media Access Control*) – e nível-3 – endereços IP. Inicialmente, o NNM tenta descobrir a rede abaixo da máscara de sub-rede em que se encontra a máquina na qual está instalado o *software*. Para isso, realiza uma captura promíscua dos pacotes da rede, contactando em seguida as máquinas cujos endereços foram capturados utilizando essencialmente ICMP (PING). Na eventualidade de resposta, confirmar-se o estado activo do nó, logo, o NNM começa a gerir esse nó, tentando obter o máximo de informação possível (usando SNMP) a intervalos regulares (*polling*). A partir deste método, o NNM consegue identificar os nós e expandir a sua descoberta para outras sub-redes, nomeadamente graças às leituras das tabelas de ARP dos equipamentos encontrados.

Se este método funciona perfeitamente para redes simples, o mesmo não acontece para redes de média/elevada dimensão com um grau de complexidade elevado. Neste caso, é necessário adicionar manualmente alguns nós principais da rede ou indicar endereços e máscaras de rede no NNM, para este efectuar uma procura mais alargada e menos restritiva da rede. Esta descoberta pode levar um tempo considerável e não conseguir gerir todos os nós automaticamente, mas não deixa de ser uma das ferramentas mais eficientes e completas de mapeamento automático de redes.

A rede mapeada é actualizada regularmente, a um dado intervalo que deve ser configurado para não sobrecarregar a rede. Esta informação é acessível num mapa com diversos níveis de detalhe que permite ter uma visão global da rede, podendo identificar, imediatamente, quais os equipamentos ligados e a funcionar correctamente, assim como os problemas existentes. Quando desejado, é possível aumentar o detalhe até ao nível das sub-redes de computadores pessoais, o que permite um controlo preciso sobre a infra-estrutura e as ligações dos equipamentos.

O NNM proporciona um conjunto de operações, tais como a procura de um equipamento na rede a partir do seu endereço MAC ou IP, identificando qual o tipo de equipamento em questão e qual a sua localização. É então possível localizar e controlar todo o equipamento existente numa rede, desde que previamente detectado pelo NNM.

Por outro lado, os agentes podem ser configurados para enviarem interrupções para a máquina onde se encontra o NMS. O NNM proporciona a criação e organização de alarmes, tanto para avisar o administrador de rede responsável da ocorrência, como para executar mecanismos de reposta automática, programados antecipadamente.

Concluindo, o NNM é uma ferramenta que poderia vir de encontro a alguns requisitos deste projecto, mas apresenta a desvantagem de ser um *software* muito dispendioso, podendo necessitar de investimentos em módulos ou outros *plugins* para poder funcionar correctamente numa rede. A informação é proprietária e nem sempre facilmente exportável, o que torna as opções de gestão limitadas às operações efectuadas pelo NNM, não sendo directamente focadas na rede do IST. A possibilidade de migração dos dados para bases de dados comerciais também apresenta uma barreira devido ao facto de este ser um projecto académico e não justificar um grande investimento em *software* paralelo para o efeito. Contudo, nos primeiros meses, esta começou por ser a ferramenta escolhida para o projecto, visando nomeadamente um aprofundamento do conhecimento sobre as suas potencialidades. No entanto, dado o custo da licença e as limitações anteriormente mencionadas, optou-se pelo desenvolvimento de uma ferramenta específica para a monitorização da rede do IST, como se descreve na próxima secção.

2.3.4. Ferramentas Dedicadas

A alternativa ao uso das ferramentas estudadas passa pela utilização das tecnologias existentes, tal como o ICMP ou o SNMP, para se criar uma nova plataforma de gestão. É possível desenvolverem-se aplicações que realizem *polling* e a descoberta da rede, do mesmo modo que as plataformas existentes. Todavia, uma ferramenta, criada de raiz e a par de uma rede específica, consegue ser mais orientada para as necessidades da mesma, sendo mais flexível e adaptada aos requisitos locais de monitorização e gestão. A informação pode ser guardada numa base de dados dinâmica, criada para o efeito, na qual se podem realizar pesquisas e correlações de dados, visando a obtenção de informação sobre a infra-estrutura da rede ou sobre o seu funcionamento. É então possível efectuar todo o tipo de operações, tanto na rede como na base de dados, permitindo tarefas de monitorização e gestão, tendo sempre em conta a possibilidade de expansão e agregação com outras ferramentas, externas ou complementares a este trabalho.

Uma plataforma deste tipo pode ser desenvolvida adoptando quer uma linguagem compilada convencional, quer uma linguagem de *scripting*. Neste trabalho, adoptou-se a segunda opção devido à legibilidade do código e à simplicidade com que se podem realizar alterações ou acrescentos, tendo em conta, como já referido anteriormente, que a flexibilidade é um dos principais benefícios deste tipo de solução. Uma das desvantagens óbvias desta opção corresponde ao seu fraco desempenho relativamente a um executável compilado e optimizado. No entanto, este factor é pouco significativo, pois o tempo de execução do programa é desprezável face à demora da recolha de informação na rede.

Várias linguagens de *scripting* são possíveis, tais como Python, Perl ou Shell. Neste trabalho, opta-se pelo Python [LuAs03] devido à sua simplicidade de sintaxe e pela quantidade de funcionalidades e bibliotecas já implementadas. Destacam-se o Pysmp [Pysn06], biblioteca que possibilita o envio e recepção dos comandos de SNMP necessários, tais como o GET ou o SET, e o Mysqldb [Mysq06], permitindo o acesso concorrente à base de dados MySQL, tanto para escrita como para leitura.

Esta solução permite a obtenção de uma ferramenta personalizada, moldável às necessidades e facilmente expansível a novos equipamentos ou configurações no futuro. As funcionalidades não são tão completas e diversificadas como as plataformas comerciais, mas são totalmente configuráveis e adaptadas à rede em questão, neste caso a rede do IST, podendo mesmo vir a ser mais eficientes e eficazes nalguns aspectos.

3. Descoberta da Topologia de Rede

3.1. Introdução

Um conhecimento sobre a topologia de rede é determinante para inúmeras tarefas de gestão, tais como a correlação de eventos ou a localização da origem de um problema. Se existir um sistema de alarmes numa rede IP centralizada, por exemplo, um simples alarme nos ramos principais pode originar uma rajada de outros, em diferentes pontos da rede, bastando para tal que os sistemas estejam interligados ou dependam uns dos outros. Muitas vezes torna-se necessária uma longa pesquisa, equipamento a equipamento, até se localizar a raiz do sucedido e então poder iniciar diligências para a sua resolução. O conhecimento da topologia é neste caso fulcral para filtrar e identificar a verdadeira causa do alarme, evitando assim a procura exaustiva. Esta informação pode ser utilizada não só na administração de redes, permitindo uma mais rápida e eficaz reacção, identificação e prevenção dos problemas, mas também na planificação de novas configurações e expansões.

Devido à complexidade e ao dinamismo das redes de hoje, obter a topologia manualmente é uma tarefa demorada e trabalhosa que tem de ser repetida regularmente devido às alterações sofridas pela rede. Tal como referido anteriormente, a rede do IST é gerida essencialmente de uma forma descentralizada, por vários gestores locais que nem sempre são informados, em tempo útil, de todas as alterações efectuadas. Esta situação é agravada pelo facto de se utilizar no IST, assim como em grande parte das redes actuais, o *Spanning Tree Protocol* (STP). Este algoritmo, definido em IEEE 802.1D [IEEE04], permite a redundância de caminhos e a tolerância a falhas na rede mas provoca, por consequência, um aumento significativo no número de alterações na topologia.

As redes contêm frequentemente serviços e sistemas críticos cujo funcionamento tem de ser assegurado ininterruptamente. Devido à existência de falhas e anomalias, a continuidade só é possível se existir redundância na rede, possibilitando assim uma alternativa ao equipamento em falta. Torna-se vital a configuração de diversos caminhos alternativos para cada equipamento possibilitando a comunicação entre nós interligados, mesmo que uma das ligações principais falhe.

Todavia, este sistema provoca ciclos no encaminhamento de pacotes, já que existem vários caminhos para o mesmo destino. Os pacotes podem voltar a passar pelos mesmos nós e repetir o trajecto indefinidamente, impedindo o fluxo de informação. Para resolver esta situação, os *switches* utilizam frequentemente o protocolo STP que permite a redundância de caminhos, sem deixar de prevenir ciclos indesejáveis na rede. Este algoritmo é dinâmico: os *switches* escolhem os caminhos para as ligações em tempo real, consoante o estado da rede, trocando para tal mensagens uns com os outros. Como resultado, verificam-se mudanças de topologia automáticas, sem acção nem

conhecimento directo dos gestores de rede. Pode então concluir-se que uma ferramenta que permita a descoberta automática da estrutura e das ligações de rede é uma forte ajuda nas tarefas de monitorização e gestão.

A descoberta de nível 3 (camada de rede) é relativamente simples de realizar, bastando para tal utilizar a informação contida nas tabelas de encaminhamento dos *routers*, dado estes necessitarem de conhecer a localização dos seus vizinhos para cumprirem a sua função. Utilitários tais como o *ping* ou o *traceroute* e objectos das MIBs obtidos por SNMP são usados por diversas ferramentas para obter a topologia de nível 3. Tem-se o exemplo de [Stot02b], cuja base consiste na obtenção da informação de conectividade ao nível de uma rede IP, e de [LWWC01], onde o foco é dirigido à topologia distribuída ao nível da Internet, ligando diferentes redes.

No entanto, a topologia de nível 3 contém apenas uma pequena parte das interligações de uma rede pelo facto de não contemplar o equipamento de nível 2 (camada de dados) tais como os *switches* e as *bridges*. A descoberta deste tipo de ligações é menos evidente que no caso anterior porque um *switch* não tem necessariamente conhecimento dos outros *switches* da rede. Um *switch* pode conhecer a localização dos seus vizinhos se este tiver STP activado, mas pode também estar directamente ligado a um *router*, não contendo assim conhecimento sobre nenhum vizinho do seu tipo, logo de nenhuma informação sobre a topologia da rede em que se encontra.

No ano 2000, a IETF propôs um RFC sobre uma MIB de topologia física [BiJo00], que tinha por objectivo resolver os problemas de aquisição da topologia para níveis abaixo do nível 3. Todavia, este RFC apenas define os objectos dessa MIB e falha na apresentação de mecanismos para os obter. A Cisco, a Intel e outras empresas fornecedoras de *hardware* desenvolveram protocolos proprietários de descoberta de topologia mas estes são de pouca utilidade numa rede heterogénea com equipamentos de diversas marcas e características. Existem algumas plataformas comerciais que reclamam permitir uma descoberta automática da rede, tais como o HP Open-View, Peregrine's InfraTools ouBindView's NETinventory. No entanto, estes são *softwares* dispendiosos, a informação resultante é pouco exportável e todos necessitam de um demorado e complexo processo de configuração antes de fornecerem quaisquer resultados satisfatórios. Em alternativa, optou-se pelo desenvolvimento de uma ferramenta dedicada, flexível e adaptada às necessidades, permitindo a realização de uma descoberta da topologia de rede, assim como diversas operações e tarefas de gestão.

3.2. Estado da Arte

3.2.1. Trabalhos Relacionados

Foram criados diversos algoritmos focados na descoberta da topologia de nível 2, utilizando diferentes métodos e técnicas. Alguns, mais simples, necessitam de fortes requisitos para o seu funcionamento, limitando assim a sua adaptabilidade às redes reais. Como resposta, são frequentemente utilizados outros métodos, mais complexos, fazendo uso de heurísticas para se aproximarem dos resultados esperados, adaptando-se assim às dificuldades encontradas nas redes actuais. Neste capítulo vão estudar-se as principais técnicas existentes actualmente, assim como as suas aplicações à rede do IST.

Se uma rede utilizar o STP, é possível obter alguma informação sobre o desenrolar deste protocolo consultando as MIBs correspondentes por SNMP. Para um dado *switch*, a partir dos objectos definidos em MIB-2.do1dBridge.dot1dStp, é possível obterem-se os resultados do STP e, assim, inferir a hierarquia dos *switches* adjacentes. Esta possibilidade deve-se ao facto do STP necessitar de informação sobre os seus vizinhos para ser executado, levando os equipamentos a comunicarem uns com os outros para descobrirem as suas inter-ligações. Este método é utilizado em [Stot02a] e analisado em [LPZL05], mas os resultados obtidos não são nem robustos nem satisfatórios para uma rede complexa. O sucedido deve-se ao facto de nem todo o equipamento dispor de STP activo e, mesmo quando utilizado, a topologia obtida corresponde apenas a parte da rede, não se tendo uma visão global da mesma. Em termos de implementação, é necessário um estudo prévio dado o formato dos objectos da MIB-2 variar de fabricante para fabricante, dificultando também a expansão a novos equipamentos. Sendo a rede do IST heterogénea e complexa, este método não é aconselhado e, como tal, não irá ser aplicado neste trabalho.

Uma abordagem diferente consiste em observar o tráfego em cada porto dos equipamentos, informação que pode ser obtida por SNMP, para em seguida ligar os portos cujo tráfego é o mais semelhante, mediante as amostras recolhidas. Este método permite a detecção e distinção de equipamentos inalcançáveis por SNMP, tais como *hubs* ou *switches* sem gestão, desde que o equipamento circundante seja monitorizado. No entanto, este algoritmo é lento na sua execução e necessita de um elevado tempo de processamento para produzir resultados, sendo estes no geral pouco animadores para redes de dimensão considerável. Para melhorar a descoberta, em [LPZL05], tenta-se adicionar outro tipo de informação às heurísticas, nomeadamente sobre o encaminhamento de pacotes, mas os resultados continuam limitados e muito dependentes de aproximações.

Para uma rede em que o equipamento não suporta em grande parte SNMP, existe a alternativa

de injectar pacotes como sondas e observar onde são entregues. Em [BDFo04], a injeção de pacotes marcados, tal como o registo dos pacotes recebidos, é realizada por uma aplicação que necessita ser instalada na maioria dos equipamentos. Este método está dimensionado para uma pequena rede, como uma rede doméstica ou a de um laboratório, mas é inconcebível numa rede de média/grande dimensão, para a qual se torna impraticável instalar e correr uma aplicação em todos os equipamentos existentes.

3.2.2. Metodologias Seleccionadas

Um dos métodos mais estudados e utilizados actualmente consiste em inferir a topologia a partir da informação contida nas tabelas de encaminhamento dos *switches*, as *Address Forwarding Table* (AFT). Uma AFT de um *switch* consiste numa tabela em que se indicam os endereços MAC dos equipamentos a ele ligados, assim como o porto respectivo em que estes foram detectados. A partir do cruzamento das AFTs de vários *switches*, torna-se possível deduzir a existência de ligações entre eles, o que irá ser demonstrado posteriormente neste trabalho.

Esta informação pode ser acedida por SNMP e, estando localizada na MIB-2, pode ser obtida pela maioria dos equipamentos, o que torna esta metodologia exportável à grande parte das redes de hoje. Variados algoritmos baseados nas AFTs foram implementados, sendo as diferenças devidas essencialmente às heurísticas utilizadas na procura de ligações.

A técnica mais simples consiste em impor requisitos mínimos de informação, apenas verificados se as AFTs contiverem todos os endereços dos equipamentos de rede. Este método foi apurado em [BGMR00] e implementado em [BGJM04] com bastante sucesso. Estes artigos apresentam um certo formalismo no que toca aos algoritmos, chegando mesmo a garantir resultados, mediante algumas condições iniciais. Todavia, como os *switches* mais afastados raramente comunicam entre si, os requisitos necessários são difíceis e improváveis de se obterem numa rede real. Por consequente, outras metodologias têm de ser abordadas.

Uma alternativa consiste em procurar caminhos entre *switches* na rede e, a partir destes, obter as ligações contidas nos mesmos. É o caso em [LPZL05], onde se transforma o problema de obter a topologia num problema matemático e em [BBGR03], no qual se estudam complexos métodos de procura e ordenação de caminhos. Ambos os métodos são vocacionados para redes contendo várias sub-redes, sendo, por consequência, mais pesados computacionalmente e não tão fiáveis para redes centralizadas como a do IST.

Por fim, em [LOGr01], aborda-se a descoberta de maneira oposta: obtém-se a topologia procurando a que portos de um dado *switch* um outro *switch* não pode estar ligado. Neste caso, basta um contra-exemplo para ser provada a hipótese inicial dos *switches* não se encontrarem ligados. Com as possibilidades restantes, o problema torna-se bastante mais fácil de resolver, sendo tanto mais

simples quanto maior for o número de ligações por *switch*. No caso de uma rede em estrela com um *router* no centro, como é o caso do IST, tal optimização é pouco significativa, dado a maioria dos *switches* apenas terem uma ou duas ligações com os seus semelhantes. Como tal, não será contemplado este método no seguimento deste trabalho.

Em todos os artigos, os autores defendem que os resultados dos seus algoritmos são satisfatórios para algumas redes, mediante algumas condições. Em caso algum se garante o sucesso do funcionamento para qualquer rede sem restrições.

Sendo a rede do IST complexa e composta por uma vasta gama de equipamentos heterogéneos, torna-se impossível a sua descoberta recorrendo somente aos algoritmos mais simples, necessitando de requisitos impraticáveis para cada caso. Os resultados encontrados que mais se aproximam dos pretendidos neste projecto resultam da procura e estudo de caminhos entre equipamentos. No entanto, estes métodos são direccionados a redes divididas por várias sub-redes, diferindo da matriz do IST em que todos os *switches* estão agrupados numa única rede – a VLAN (*Virtual Local Area Network*) dos *switches*. Opta-se então pelo desenvolvimento de um novo algoritmo, baseado na procura de caminhos a partir de AFTs, mas em que as diversas heurísticas criadas são originais e adaptadas a uma rede centralizada sendo, por consequente, mais eficientes e apropriadas às características do IST.

3.3. Algoritmo Realizado

3.3.1. Fundamento

O método escolhido baseia-se em obter AFTs por SNMP para, depois de tratadas, serem utilizadas na descoberta da topologia. Esta operação pode ser por vezes demorada, devido ao tempo de envio das tabelas por alguns *switches*, chegando a ultrapassar a dezena de minutos. Sendo este processo significativamente mais lento do que qualquer algoritmo envolvido, a técnica implementada foca-se em obter o máximo de informação possível das AFTs, mesmo com alguma redundância, para no final se poder inferir sobre as ligações mais prováveis e se poder concluir sobre a topologia da rede.

Alguns equipamentos não se encontram acessíveis do exterior, não se tornando possível efectuar todos os pedidos SNMP da máquina em que se realiza o trabalho. É então utilizado um túnel SSH (*Secure Shell*) para uma máquina de suporte, podendo nessa posição privilegiada efectuar as operações SNMP desejadas e assim obter toda a informação necessária, tal como as tabelas de encaminhamento dos *switches*.

Designa-se por AFT completa, a tabela de encaminhamento que contém os endereços de todos os *switches* da rede e por AFT vazia aquela que não contém nenhum deles. Vai iniciar-se o algoritmo

pela procura de ligações, tendo por requisito que todas as AFTs estejam completas, tal como previamente implementado em [BGJM04]. Sendo este requisito difícil de satisfazer na rede do IST, estuda-se em seguida uma solução que visa aumentar o preenchimento das AFT.

Um dos problemas consiste no facto dos *switches* não comunicarem forçosamente entre eles, o que resulta em AFTs muito reduzidas ou mesmo vazias. No exemplo da rede do IST, verifica-se que, na maior parte do período de aulas, o tráfego na rede é significativo, aumentando assim a frequência com que os *switches* se anunciam e utilizam protocolos de *broadcast* (como o ARP). No entanto, mesmo nesse caso, o preenchimento das tabelas de encaminhamento resultantes fica muito aquém do pretendido, não possibilitando, consequentemente, a execução do algoritmo apresentado.

Com o objectivo de forçar os *switches* a trocarem pacotes entre eles, preenchendo assim as AFTs correspondentes, explora-se o funcionamento do protocolo ARP, utilizado em cada comunicação para um destinatário não conhecido. Quando um equipamento (emissor) necessita de enviar um pacote, mas apenas possui o endereço IP do receptor, este difunde em *broadcast* um pacote ARP contendo o endereço IP do destino e espera por uma resposta com um endereço MAC respectivo. Nesta operação, o equipamento emissor anuncia-se, tendo por consequência o preenchimento das AFTs dos *switches* da rede com o seu endereço.

Para a criação e injeção de pacotes na rede, foi usado a aplicação Nemesis [GrNa04], que permite mascarar os endereços de origem e destino – *spoofing* – assim como escolher o tipo de protocolo a utilizar. Numa primeira experiência, foram criados pacotes ICMP onde os endereços de origem e de destino correspondiam a endereços de *switches* da rede, cujas AFTs se pretendiam preencher. O objectivo seria levar o destinatário do pacote a responder ao seu emissor fictício, enviando assim um pacote ARP para obter o endereço de resposta, realizando por consequência o seu anúncio à rede. Como a injeção de pacotes não é efectuada directamente na rede dos *switches*, seria de esperar que o pacote fosse encaminhado para o *router* central e, seguidamente, para a rede adequada. No entanto, o envio de um pacote ICMP com um endereço de origem de outra rede resulta num envenenamento da tabela de ARPs do *router* central. O sucedido deve-se ao facto do *router* passar a associar o endereço mascarado à interface de onde originou o pacote ICMP, que não corresponde na realidade à rede dos *switches*. O Némesis acaba assim por ser responsável por uma degradação na qualidade de serviço da rede, dado o *router* central começar a reencaminhar pacotes por um porto incorrecto. Uma possível resolução desta falha de segurança consiste em dotar o *router* central de uma funcionalidade que descarte os pacotes de rede, cujo endereço de origem provém de outra rede não associada à interface utilizada (protecção contra *spoofing*). Esta tarefa pode ser realizada, por exemplo, por uma *firewall* interna configurada com regras contra *spoofing*.

Para evitar esta situação, alterou-se o endereço IP de origem dos pacotes ICMP para um endereço não existente na rede, conseguindo-se assim evitar o envenenamento da tabela de ARP do *router*. Esta tabela passa apenas a conter uma nova entrada que não corresponde a um endereço real, não

afectando os endereços restantes. Os resultados obtidos são os esperados: o *switch*, ao receber o pacote ICMP, tenta responder ao mesmo enviando um *broadcast* de ARPs, em que se anuncia, para obter o endereço MAC do emissor fictício. Deste modo, preenche-se grande parte das tabelas de encaminhamento dos *switches* e torna-se então possível proceder à descoberta de ligações pelo algoritmo pretendido. Todas as AFTs não são totalmente preenchidas porque o protocolo ARP não garante a entrega dos pacotes, podendo alguns ficar perdidos pela rede. Adicionalmente, para redes muito congestionadas, as AFTs podem atingir a sua capacidade máxima resultando, consequentemente, na limpeza de algumas entradas na tabela antes do prazo configurado.

Verifica-se que este algoritmo produz resultados mas, como nem todas as AFTs se conseguem completar, não se torna possível inferir a topologia total da rede. Vão então ser criadas novas heurísticas baseadas na procura e ordenação de caminhos de [BBGR03], mas agora específicas a uma só sub-rede (VLAN dos *switches*), no intuito de se obter o máximo de ligações possíveis e assim se concluir sobre a topologia da rede. No desenrolar deste estudo, considera-se que as AFTs não possuem entradas inválidas, correspondendo por exemplo a ciclos na rede, situação que inviabilizaria o fluxo de informação pela rede.

Inicia-se este algoritmo com uma procura de ligações simples, apenas aplicável a AFTs completas, com o auxílio das operações de *spoofing*. Os resultados obtidos podem ser fortemente incompletos mas, se existentes, permitem ajudar na identificação de algumas ligações. Em seguida, pesquisam-se todos os caminhos potenciais entre *switches*, possibilitando resultados, mesmo no caso de algumas AFTs estarem totalmente vazias. Torna-se depois necessário ordenar os caminhos encontrados para, no final, se poderem inferir as ligações existentes nos mesmos. Por fim, estuda-se a hierarquia da rede com o objectivo de concluir sobre a posição dos equipamentos relativamente ao centro e à periferia da rede.

3.3.2. Ligações Obtidas a Partir de AFTs Completas

Começa-se por definir o formalismo utilizado, baseado na notação utilizada em [BGJM04]. Seja S_i (ou S_i) o *switch* i da rede, sendo i uma numeração arbitrária usada apenas como identificador. Cada *switch* possui um conjunto de interfaces, correspondendo aos portos físicos do equipamento. Define-se por S_{ij} a interface j do *switch* S_i . Para uma dada interface j , seja A_{ij} a AFT de S_i que corresponde a essa mesma interface e A_i o conjunto das AFTs de todas as interfaces. O conjunto de todos os *switches* da rede é representado por N .

Considera-se que dois *switches* estão directamente ligados se não existir nenhum outro *switch* no caminho entre ambos. Por outro lado, designam-se por simplesmente ligados os *switches* que estão ligados mas cujo caminho entre eles pode ou não conter outros *switches*.

Considere-se, por exemplo, a rede da Figura 3.1. Neste caso, se as AFTs estão completas, obtém-se os seguintes resultados:

- $A_{11} = \{S2, S3, S4\}$
- $A_{25} = \{S1\}$
- $A_{26} = \{S3\}$
- $A_{23} = \{S4\}$
- $A_{37} = \{S1, S2, S4\}$
- $A_{45} = \{S1, S2, S3\}$

Também é possível constatar, por exemplo, que S1 e S2 estão directamente ligados, mas que S4 e S3 apenas podem estar simplesmente ligados por terem S2 no caminho que os separa. Pode-se verificar que A_{11} só não tem *switches* comuns com A_{25} e que o conjunto destas duas AFTs contém todos os *switches* da rede. Na realidade, prova-se mais à frente, no Lema 3.1, que esta propriedade é válida para todos os casos. Se S1 e S2 apenas fossem simplesmente ligados, existiria obrigatoriamente um *switch* entre os dois e como tal A_{11} e A_{25} tê-lo-iam em comum (tal como entre A_{37} e A_{45}). Existe também a situação em que os dois portos estão localizados em direcções opostas como, por exemplo, o porto 6 de S2 e o porto 2 de S1 (não representado por não estar ligado a nenhum outro *switch*: $A_{12} = \{\}$). Neste caso, S1 e S2 não se encontram na união das duas AFTs, já que o endereço de um *switch* nunca pode estar na sua própria AFT. A demonstração formal de todas estas características segue-se à declaração do lema.

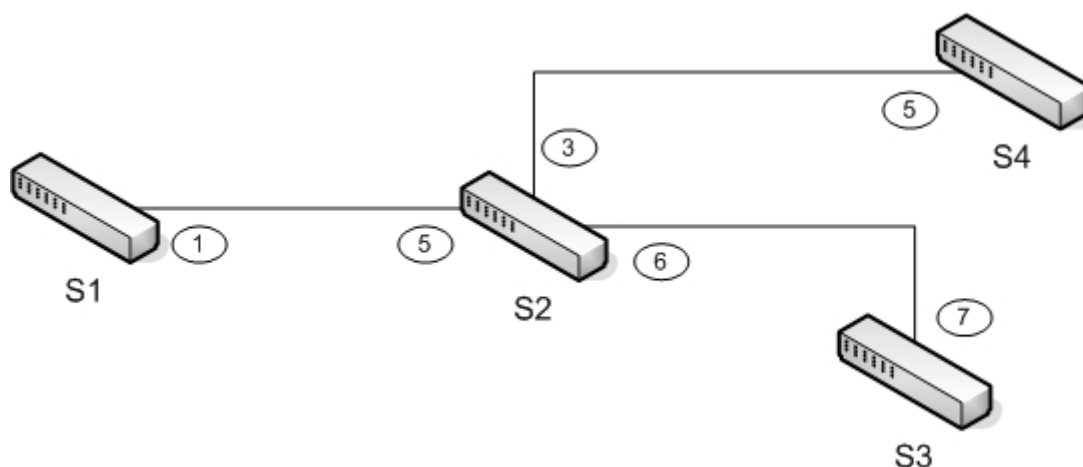


Figura 3.1 - Exemplo de uma rede contendo vários *switches*.

Considerando AFTs completas, o Lema 3.1 é suficiente para inferir todas as ligações directas existentes e, como tal, concluir sobre a topologia da rede. Este lema é uma variante do original apresentado em [BGM00], seguindo a demonstração, consequentemente, passos semelhantes.

Lema 3.1: Estando A_i e A_k completas, os *switches* S_i e S_k estão directamente ligados pelas interfaces j e l respectivamente se e só se $A_{ij} \cap A_{kl} = \emptyset$ e $A_{ij} \cup A_{kl} = N$.

Demonstração: Se as interfaces S_{ij} e S_{kl} estão directamente ligadas, então não pode existir nenhum *switch* no caminho entre elas, logo $A_{ij} \cap A_{kl} = \emptyset$. Como as AFTs estão completas, todos os *switches* da rede estão contidos na união das AFTs de duas interfaces ligadas, dado estas estarem dirigidas a partes diferentes da rede; deduz-se assim que $A_{ij} \cup A_{kl} = N$. Considera-se agora que $A_{ij} \cap A_{kl} = \emptyset$ e $A_{ij} \cup A_{kl} = N$. Sendo $A_{ij} \cup A_{kl} = N$, então as interfaces S_{ij} e S_{kl} têm de estar pelos menos simplesmente ligadas porque, no caso contrario, teríamos $S_i, S_k \notin (A_i \cup A_k)$, já que $S_i \notin A_i$ e $S_k \notin A_k$, e consequentemente $A_{ij} \cup A_{kl} \neq N$. Se S_{ij} e S_{kl} , simplesmente ligadas, não estiverem directamente ligadas, então existe um caminho entre S_{ij} e S_{kl} que contém pelos menos um *switch* e então $A_{ij} \cap A_{kl} \neq \emptyset$. Conclui-se que se $A_{ij} \cap A_{kl} = \emptyset$ e $A_{ij} \cup A_{kl} = N$, então as interfaces S_{ij} e S_{kl} têm necessariamente de estar directamente ligadas.

A Figura 3.2 mostra o pseudo-código correspondente à implementação realizada. Os resultados podem ser garantidos pelo Lema 3.1, contudo, estes necessitam sempre de AFTs completas, o que nem sempre se verifica na rede do IST. É naturalmente necessário o auxílio a outros métodos mais robustos, para poder obter os resultados pretendidos.

```

/* AFTs corresponde a uma lista contendo as AFTs de cada porto para todos os switches */
Função topologia_simples_aft_completas(AFT):
    para cada switch nas AFTs:
        para cada porto (origem):
            para cada porto (destino) dos restantes switches:
                se a intersecção das AFTs for nula e se a união das AFTs \
                corresponder a todos os switch:
                    adicionar ligação entre porto origem e destino \
                    à lista ligações_sw
                    /* ligação contém informação sobre endereços e portos */
    retornar ligações_sw

```

Figura 3.2 - Descoberta de ligações a partir de AFTs completas.

3.3.3. Pesquisa de Caminhos

Se faltar apenas um endereço à AFT de um *switch*, este último deixa de ser contemplado pelo algoritmo anterior. Como tal, para inferir qualquer tipo de ligação a esse *switch*, são necessárias outras técnicas que não requeiram de AFTs completas.

Vai iniciar-se o estudo por uma pesquisa de caminhos entre equipamentos, sendo os traços gerais semelhantes aos implementados em [BBGR03]. Todos os *switches* contidos na rede do IST encontram-se na mesma VLAN. Assim sendo, estão todos simplesmente ligados, logo, existe necessariamente um caminho entre cada par de *switches*. Este algoritmo baseia-se na procura desses mesmos caminhos, uma vez que não são necessárias AFTs completas para os poder deduzir, tal como irá ser demonstrado seguidamente.

A ideia consiste em escolher um par de *switches*, simplesmente ligados, e encontrar todos os *switches* que se localizam no caminho que os une. Um *switch* S_i apenas pode fazer parte do caminho entre dois outros *switches* S_j e S_k se S_i contiver os endereços de S_j e S_k em interfaces diferentes, logo em AFTs diferentes. Em qualquer outra situação, ambas as extremidades do caminho estão localizadas na mesma AFT (extremidades ligadas ao *switch* pelo mesmo porto) e, como tal, o *switch* encontra-se fora do caminho procurado. Este método é provado no Lema 3.2. Este lema inspira-se nos trabalhos de [BBGR03] mas foi concebido especificamente para este trabalho. A demonstração que se segue garante que a sua utilização é eficaz.

Lema 3.2: Sejam S_i , S_k e S_m três *switches* simplesmente ligados. Sendo C o caminho que une S_i a S_k ; S_m faz parte de C se $S_i \subset A_{mr}$ e $S_k \subset A_{ms}$, com $r \neq s$.

Demonstração: Se S_i , S_k e S_m estiverem simplesmente ligados e $S_i, S_k \subset A_m$, então existem apenas 4 situações possíveis:

1. S_m ligado a S_i , que está ele mesmo ligado a S_k a partir de outra interface, S_m localizando-se assim fora de C : $S_i, S_k \subset A_{mr}$, ou seja S_i e S_k encontram-se na mesma interface de A_m .
2. S_m ligado a S_k , que está ele mesmo ligado a S_i a partir de outra interface, S_m localizando-se assim fora de C : $S_i, S_k \subset A_{ms}$, tal como no caso anterior.
3. S_m ligado entre S_i e S_k , não fazendo parte de C (esta situação ocorre, por exemplo, quando os três *switches* estão directamente ligados a um mesmo *Hub*): A_i contém S_m e S_k na mesma interface, assim como A_k com S_m e S_i ; mas tanto S_i como S_k encontram-se na mesma interface de A_m ($S_i, S_k \subset A_{ms}$).
4. S_m ligado entre S_i e S_k , fazendo parte de C : mesmas condições que no ponto 3 excepto para S_m que agora passa a ter $S_i \subset A_{mr}$ e $S_k \subset A_{ms}$, com $r \neq s$.

Conclui-se que para S_m fazer parte de C , apenas é necessário que $S_i \subset A_{mr}$ e $S_k \subset A_{ms}$, com $r \neq s$.

O recíproco do Lema 3.2 não é válido, dado as tabelas AFTs poderem não ser completas: é possível S_m fazer parte do caminho sem que o Lema 3.2 se verifique, bastando para tal que A_m não contenha S_i ou S_k . Por outro lado, estando as AFTs incompletas, é possível que os caminhos encontrados não contenham todos os *switches* que se localizem realmente nesse caminho. Desse modo, não se pode garantir que os caminhos encontrados correspondam a ligações directas entre *switches*, dedução que só poderá ser aproximada a partir de heurísticas.

Se, por exemplo, não for encontrado nenhum *switch* no caminho C entre um par de *switches*, C é descartado já que este não fornece nenhuma informação adicional. No caso extremo de todas as AFTs estarem vazias, os caminhos encontrados são todos descartados já que apenas são compostos por dois *switches*, processo que se repete para todos os pares de *switches* possíveis, o que não corresponde obviamente à realidade física das ligações.

Todavia, tomando como exemplo a Figura 3.1, para encontrar o caminho entre S_1 e S_3 , apenas necessitamos que $S_1, S_3 \subset A_2$, o que diminui fortemente os requisitos necessários ao algoritmo

implementado a partir do Lema 3.1. A partir de algumas AFTs, torna-se possível obter informação, mesmo que ainda não organizada e utilizável, permitindo encontrar caminhos cujos extremos correspondem a *switches* contendo AFTs vazias.

Este método tende a ter resultados satisfatórios pelo facto dos *switches* comunicarem mais provavelmente com os vizinhos mais próximos do que com aqueles na extremidade oposta da rede, devido ao facto da rede do IST de se tratar de uma rede em estrela. A maior parte dos pacotes é distribuída pelo centro, o que resulta no preenchimento das AFTs necessárias para a obtenção de caminhos entre diversos pontos da rede e o centro.

Nesta fase, não é levada em conta a ordem pela qual os *switches* se encontram no caminho, problema que irá ser resolvido no capítulo seguinte. São excluídos os caminhos exactamente iguais, em que apenas a ordem dos equipamentos varia, porque o resultado do ordenamento seria idêntico. A filtragem de caminhos redundantes não é, no entanto, realizada nesta etapa, pois a mesma só pode ser efectuada correctamente a partir de caminhos ordenados. Podem existir caminhos com mais detalhe do que outros, permitindo obter mais e diferente informação depois de ordenados. Por outro lado, caminhos não directamente ordenáveis com as AFTs disponíveis podem vir a ser descobertos graças à integração da informação de outros caminhos mais pequenos, parcialmente sobrepostos aos primeiros, que consigam ser ordenados.

O pseudo-código desta procura de caminhos encontra-se na Figura 3.3.

```
Função pesquisa_caminhos_sw(AFT):  
  para cada par de switches S1 e S2 tirados de AFT:  
    seja C o caminho entre S1 e S2  
    para cada switch S3 ≠ S1 e S2:  
      se S3 tiver S1 e S2 em duas interfaces diferentes da sua AFT:  
        adicionar S3 a C  
      se C for constituído por mais de 2 switches e não corresponder a um caminho já \  
      existente:  
        adicionar C à lista caminhos_ao_ordenados  
  retornar caminhos_ao_ordenados
```

Figura 3.3 – Procura de caminhos a partir de AFTs potencialmente incompletas.

3.3.4. Ordenação de Caminhos

Para poder utilizar a informação contida nos caminhos, é necessário que estes estejam correctamente ordenados. Entende-se por caminho ordenado aquele cuja ordem dos equipamentos entre as extremidades do caminho corresponde à ordem real pela qual são percorridos se um pacote for enviado de uma extremidade para a outra. A dificuldade reside no facto das AFTs serem incompletas, não permitindo sempre o ordenamento directo dos *switches*. Vão-se utilizar diversas heurísticas para procurar obter o maior número possível de caminhos ordenados a partir dos dados existentes.

Começam-se por destacar os caminhos com apenas três *switches* dado estes já se encontrarem naturalmente ordenados. Nos restantes casos, definem-se por E1 e E2 os *switches* das extremidades a partir das quais se formou um caminho. O primeiro objectivo consiste em utilizar as ligações obtidas a partir de AFTs completas, se existentes, para ordenar o equipamento encontrado. Sendo as ligações garantidas pelo Lema 3.1, estas permitem concluir directamente sobre o ordenamento, sem qualquer verificação adicional. Como as cadeias de ligações são iniciadas nas extremidades e apenas existem ligações directas (Lema 3.1), em qualquer situação em que apenas reste um *switch* por ordenar, pode concluir-se o processo, pois a localização do *switch* é obtida por exclusão de partes. Se, por exemplo, num caminho com dois *switches* por ordenar se identificar uma ligação entre um deles e uma das extremidades do caminho, então é imediatamente possível concluir sobre o ordenamento, sendo o último *switch* localizado na posição restante. Na Figura 3.4, podem-se observar os passos desta primeira operação.

```

Função ordena_caminhos(AFT, ligações_sw, caminhos):
  /* ligações_sw corresponde às ligações obtidas a partir de AFTs completas */
  para cada caminho C em caminhos:
    sejam E1 e E2 as extremidades de C
    seja D a lista de switches desordenados entre E1 e E2
    enquanto forem ordenados switches:
      para cada ligação L em ligações_sw:
        para cada switch S de D:
          se L ligar E1 a S:
            retirar S de D
            E1 = S
          senão, se L ligar E2 a S:
            retirar S de D
            E2 = S
    se apenas resta um switch -> ordená-lo por exclusão de partes
    se C não se encontra completamente ordenado
      /* tentar ordenar o que falta a partir das AFTs – ver próximo parágrafo */
      ordena_caminho_com_aft(AFT,C,E1,E2, caminho_parcialmente_ordenado)
      actualiza C
    se C completamente ordenado:
      procurar portos das extremidades do caminho se desconhecidos
      actualizar C
      adicionar C à lista caminhos_ordenados
    senão:
      não se pode concluir logo descarta-se C.
  retornar caminhos_ordenados

```

Figura 3.4 – Ordenamento de caminhos com base em ligações garantidas.

Se este método não for suficiente para ordenar o caminho, recorre-se às AFTs para pesquisar as ligações em falta. Inicia-se o processo escolhendo um *switch* P da lista de desordenados que se encontra naturalmente posicionado entre E1 e E2. Em seguida, para cada *switch* S restante, utiliza-se o Lema 3.2 para verificar se S se localiza em algum caminho entre todos os pares possíveis de *switches* ordenados (E1, E2 e P). Se encontrado, S passa a estar ordenado entre os dois *switches* correspondentes. Na eventualidade das AFTs de S não conterem informação suficiente para especificar a sua localização, repetir-se-á a procura utilizando as AFTs dos *switch* já parcialmente

ordenados (E1, E2 e P), tornando possível, mais uma vez, inferir a localização de S mesmo que este contenha informação muito incompleta nas suas AFTs. Seguidamente, este método é repetido para todos os *switches* desordenados. Neste caso é mesmo necessário ordenar todos os *switches*: como se tratam de ligações simples, o último *switch* não se encontra automaticamente ordenado, tal como acontecia com as AFTs completas (ligações directas). O algoritmo implementado pode ser estudado com um maior detalhe graças à Figura 3.5.

Nos casos em que se recorre ao método de ordenamento por AFTs, frequentemente os portos dos *switches* das extremidades E1 e E2 não são conhecidos. Nesta circunstância, é possível tentar obter esse conhecimento procurando nas AFTs de E1 e E2 por qualquer equipamento localizado no caminho observando, em caso de sucesso, a interface utilizada. No caso de não ser encontrado, resta a possibilidade de procurar os endereços MAC de E1 e E2 nas AFTs dos *switches* existentes no caminho. Este endereço varia frequentemente de porto para porto tornando possível deduzir qual a interface em questão. Não deixam de existir situações em que as interfaces das extremidades permanecem não identificadas levando, nesses casos, a informação contida no caminho a não ser totalmente utilizável, como irá ser demonstrado no próximo capítulo.

```

Função ordena_caminho_com_aft(AFT,C,E1,E2,caminho_parcialmente_ordenado):
    seja P a lista de switches ordenados do caminho entre E1 e E2
    adicionar o primeiro switch de C entre E1 e E2 a P
    para cada switch S de C entre E1 e E2, S não contido em P:
        procurar posição relativamente aos outros switches de P, usando para isso \
        a AFT de S tentando localizar S em caminhos auxiliares entre pares de P

        se a localização não for encontrada, criar caminhos auxiliares entre S e \
        cada switch de P para tentar localizar os restantes switches de P nesse \
        caminho, usando agora as AFTs dos switches de P

        se S não tiver sido ordenado:
            função retorna um valor nulo, indicando a exclusão de C
    /* Se todos os switches do caminho forem ordenados */
    construir caminho ordenado juntando toda a informação recolhida
    retornar caminho_ordenado

```

Figura 3.5 – Ordenamento de caminhos com base em AFTs.

Tendo como ponto de partida a Figura 3.6, exemplificam-se agora mais claramente algumas das funcionalidades propostas pelo algoritmo. Consideram-se as seguintes AFTs (incompletas):

- $A_{11} = \{ \}$
- $A_{25} = \{S1\}$
- $A_{26} = \{S3,S5\}$
- $A_{37} = \{S1 \}$
- $A_{31} = \{S4,S5\}$
- $A_{42} = \{S1\}$
- $A_{43} = \{S5\}$
- $A_{52} = \{S4 \}$

Escolhe-se a pesquisa do caminho entre S1 e S5. O caminho encontrado contém os *switches* S3, S2 e S4, dado todos estes equipamentos conterem os endereços de S1 e S2 em portos diferentes. Apresenta-se seguidamente, em detalhe, os passos do algoritmo de ordenamento:

- Seja C o caminho que se deseja ordenar e O o caminho parcialmente ordenado.
- Inicialmente $C = \{S3, S2, S4\}$ e $O = \{\}$.
- Começa-se por escolher um *switch* para fazer parte do caminho: $O = \{S3\}$, $C = \{S2, S4\}$. Qualquer *switch* é possível já que todos se encontram entre S1 e S5 e um caminho com três *switch* está naturalmente ordenado.
- Em seguida, retira-se um novo *switch* a C: S2.
- Procura-se pela localização de S2 usando A_2 . Como S1 e S3 estão presentes em portos diferentes de A_2 , a posição de S2 foi encontrada entre S1 e S3: $O = \{S2, S3\}$, $C = \{S4\}$.
- Retira-se o último *switch* de C: S4.
- Procura-se pela localização de S4 usando A_4 . No entanto, não foi possível colocar S4 entre os pares (S1,S2), (S2,S3) e (S3,S5), já que A_4 está fortemente incompleta.
- Procura-se pela localização de S4 usando A_2 : S4 não se encontra presente
- Procura-se pela localização de S4 usando A_3 : S4 foi encontrado na mesma interface que S5. Estando S4 localizado necessariamente entre S1 e S5, este só se pode encontrar entre S3 e S5. É então possível ordenar S4.
- $C=\{\}$ logo retorna-se o caminho O: $\{S2, S3, S4\}$

Verifica-se que, mesmo a partir de AFTs pouco completas (A_1 vazia), foi possível ordenar o caminho e assim obter alguma informação sobre a topologia da rede.

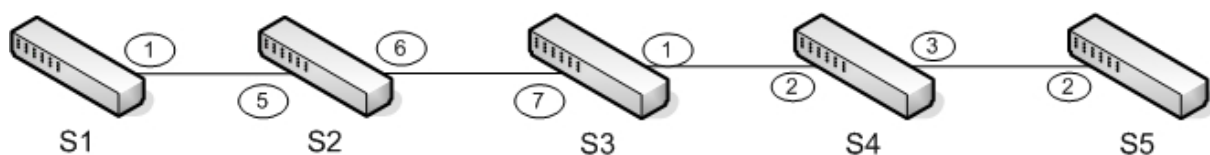


Figura 3.6 – Exemplo de uma cascata de *switches*.

3.3.5. Dedução de Ligações Directas

Os resultados obtidos a partir de AFTs completas são os únicos que garantem ligações directas entre *switches*. Os caminhos apenas indicam ligações simples, existindo sempre a possibilidade destas não conterem todos os *switches* reais na rede. No entanto, graças às técnicas de *spoofing* ou a volumes

de tráfego importante, os resultados das heurísticas aproximam-se fortemente dos reais, como se poderá verificar na apresentação de resultados, na parte final deste relatório.

É necessário proceder a uma filtragem dos caminhos para retirar a informação redundante ou contraditória. Cada par de *switches* consecutivos, num caminho, vai fornecer uma potencial ligação. Esta só é validada se não existir nenhum outro caminho em que este par de *switches* é separado por outro *switch*. Esta heurística pode ser evidenciada pela Figura 3.7. No caso da Figura 3.6, considera-se, por exemplo, que o algoritmo encontra os caminhos C1 e C2 correspondendo a {S1,S2,S4} e a {S2,S3,S4} respectivamente. As ligações correctas correspondem a S1-S2, S2-S3 e S3-S4 mas não a S2-S4 como seria sugerido por C1. Mostra-se, no exemplo anterior, que a validação de uma ligação é essencial para garantir que esta possa ter alguma semelhança com a realidade.

Com o elevado número e redundância de caminhos, este algoritmo tende a convergir para as ligações directas da rede excluindo, obviamente, todo e qualquer *switch* não encontrado em nenhuma AFT. Resta apenas identificar a hierarquia da rede para se poder concluir sobre a topologia existente.

```
Função infere_ligacoes(ligações_sw,caminhos_ordenados):
    _ligações_ = lista contendo ligações_sw
    para cada caminho C em caminhos_encontrados:
        para cada par de switches S1 e S2 consecutivos no caminho:
            se a ligação entre S1 e S2 já fizer parte de _ligações_ :
                descartar a ligação S1-S2 deste caminho
            senão, para cada caminho:
                se forem encontrados 3 switches consecutivos em que os \
                switches das extremidades correspondem a S1 e S2:
                    a ligação S1-S2 não é válida: esta é descartada
            se ligação não tiver sido descartada, adicionar S1-S2 à lista _ligações_
    retornar _ligações_
```

Figura 3.7 – Dedução das ligações entre *switches* a partir dos caminhos.

3.3.6. Hierarquia do Equipamento

Numa rede em estrela, a hierarquia pode ser definida como a distância do equipamento ao centro da rede, em que a distância corresponde ao número de ligações entre *switches* no trajecto. Na rede do IST, configurada em estrela, todo o tráfego direccionado para fora, assim como grande parte do tráfego interno, passa obrigatoriamente pelo *router* central. Define-se então por porto de saída, aquele a partir do qual um *switch* se liga ao centro. Nesse porto acede-se à grande maioria da rede, sendo apenas excluídos os equipamentos mais baixos na hierarquia.

Esta informação é relevante, nomeadamente para qualquer tarefa de localização. Permite, por exemplo, responder a questões como a seguinte: se existir um caminho S1-S2-centro e se o endereço MAC de um computador pessoal (PC) se encontrar nas AFTs de S1 e de S2, como saber a qual destes *switches* o PC se encontra efectivamente ligado? Conhecendo a hierarquia, pode-se

facilmente constatar que apenas dois casos são possíveis: se o PC estiver ligado directamente a S2, então este encontra-se em A_{1x} , x correspondendo ao porto de saída de S1; no caso contrário, o PC encontra-se directamente ligado a S1, logo presente em A_{1y} , sendo y um porto necessariamente diferente de x .

Partindo do facto do conhecimento da hierarquia dos equipamentos ser o resultado final da descoberta da topologia, inicialmente foi tentado inferir directamente essa informação a partir das AFTs. Implementou-se um algoritmo que procurasse por ligações entre pares de *switches* S_i , S_j para os quais $A_i \subset A_{jx}$, ou seja, A_j contendo todos os endereços de A_i numa única interface. Esta situação corresponde a S_i ser inferior, em termos de hierarquia, e estar directamente ligado a S_j pelo porto de saída. No entanto, para se obterem resultados numa rede real, é necessário parametrizar um nível de tolerância já que nem todos os endereços de A_i se encontram em A_j , devido ao facto de nem todos os pacotes circulando por S_i passarem por S_j e nem todas as AFTs estarem completas. A única solução passa por pesquisar apenas uma fracção de endereços de A_i numa interface de A_j . Infelizmente, tal metodologia introduz um número significativo de falsos positivos, mesmo antes de se conseguirem encontrar todas as ligações reais. Consequentemente, esta técnica foi abandonada, vindo a procura de hierarquia a ser executada apenas depois de se conhecerem as ligações na rede.

Por último, é necessário ter em conta a importância dos portos das ligações serem conhecidos. As ligações, obtidas pelo algoritmo de análise de caminhos, apenas podem ser de dois tipos. Uma ligação, entre dois *switches*, que é efectuada pelos portos de saída de ambos: estes não estão fisicamente conectados um ao outro, dado o *router* central se encontrar necessariamente no meio deles. Ou então, uma ligação entre um *switch*, mais baixo na hierarquia, que se liga pelo porto de saída, e um outro, cuja interface utilizada na ligação é diferente da de saída, que se encontra mais próximo do centro. Não é possível ter uma ligação entre *switches*, em que ambos os portos são diferentes dos de saída, dado isso significar a existência de um ciclo na rede: o tráfego entre os dois *switches* em questão poderia ser encaminhado pela dita ligação ou pelo centro da rede, o que provocaria falhas graves no funcionamento global.

Conclui-se que, se ambos os portos de uma ligação forem desconhecidos, esta tem de ser excluída, não havendo forma de se saber se o *router* se encontra no meio desta ou não. Nos casos em que apenas um dos portos não é conhecido, existe a possibilidade de deduzir a hierarquia: se o outro porto não for de saída, o porto desconhecido é-lo obrigatoriamente (senão existiria um ciclo na rede) e a ligação obtida é hierárquica. Em todos as outras situações, o *router* pode, ou não, estar localizado no meio da ligação, não havendo possibilidade de se inferir qualquer outro tipo de informação. A ligação é nesses casos excluída. Todos estes passos podem ser aferidos no pseudo-código da Figura 3.8.

```

Função obter_hierarquia (AFT, _ligações_):
  para cada ligação L em _ligações_:
    se ambos os portos forem desconhecidos:
      não se pode concluir -> descarta-se essa ligação
    senão, se um porto for desconhecido:
      se o outro não for porto de saída:
        adiciona-se a ligação hierárquica, em que o porto desconhecido \
        corresponde ao porto de saída, à lista ligações_finais
      no caso contrario:
        /* o router pode ou não encontrar-se no meio da ligação */
        não se pode concluir -> descarta-se essa ligação
    senão, se ambos os portos forem os portos de saída:
      /* o centro da rede, router, encontra-se no meio da ligação */
      adicionam-se 2 ligações switch-router à lista ligações_finais, \
      correspondendo a cada um dos switches de L
    senão:
      criar ligação hierárquica sendo o switch mais baixo na hierarquia \
      aquele que usa, nesta ligação, o seu porto de saída
      adicionar ligação à lista ligações_finais
  retornar ligações_finais

```

Figura 3.8 – Dedução da hierarquia de rede.

Observando o exemplo da Figura 3.9, vai-se procurar ordenar os *switches* em termos de hierarquia *filho-pai* (sendo o *filho* o *switch* mais afastado do *router*), pondo em evidência todas as conclusões estudadas:

- Seja P_i , o porto de saída de S_i .
- Ligação S1-S2: ambos os portos são conhecidos; $S_{11} = P_1$, mas $S_{25} \neq P_2$, logo S1 corresponde ao *switch* *filho*.
- Ligação S2-S4: $S_{26} = P_2$ e $S_{41} = P_4$, logo são deduzidas duas ligações: S2-*router*, em que S2 é o *filho*, e S4-*router*, em que S4 é o *filho*.
- Ligação S4-S5: $S_{42} \neq P_4$, logo o porto de S5 só pode ser o de saída e consequentemente S5 só pode corresponder ao *switch* *filho*.
- Ligação S4-S6: $S_{41} = P_4$, mas neste caso, o porto de S6 é desconhecido; não se pode então concluir sobre a ligação.
- Ligação S3-S6: $S_3 = P_3$, mas, tal como no caso anterior, não sendo o porto de S6 conhecido, não se pode concluir. Uma ligação contendo o *router* pelo meio (S4-S6) pode conter o mesmo tipo de informação que um ligação directa (S3-S6), não havendo possibilidade de as distinguir.

A partir deste algoritmo, ajudado pelas heurísticas, obtém-se informação importante sobre a topologia da rede. Sendo esta baseada em AFTs incompletas, não é possível garantir que todas as ligações tenham sido efectivamente encontradas. No entanto, os resultados obtidos, sendo coerentes, facultam desde já uma preciosa ajuda em operações de monitorização e gestão.

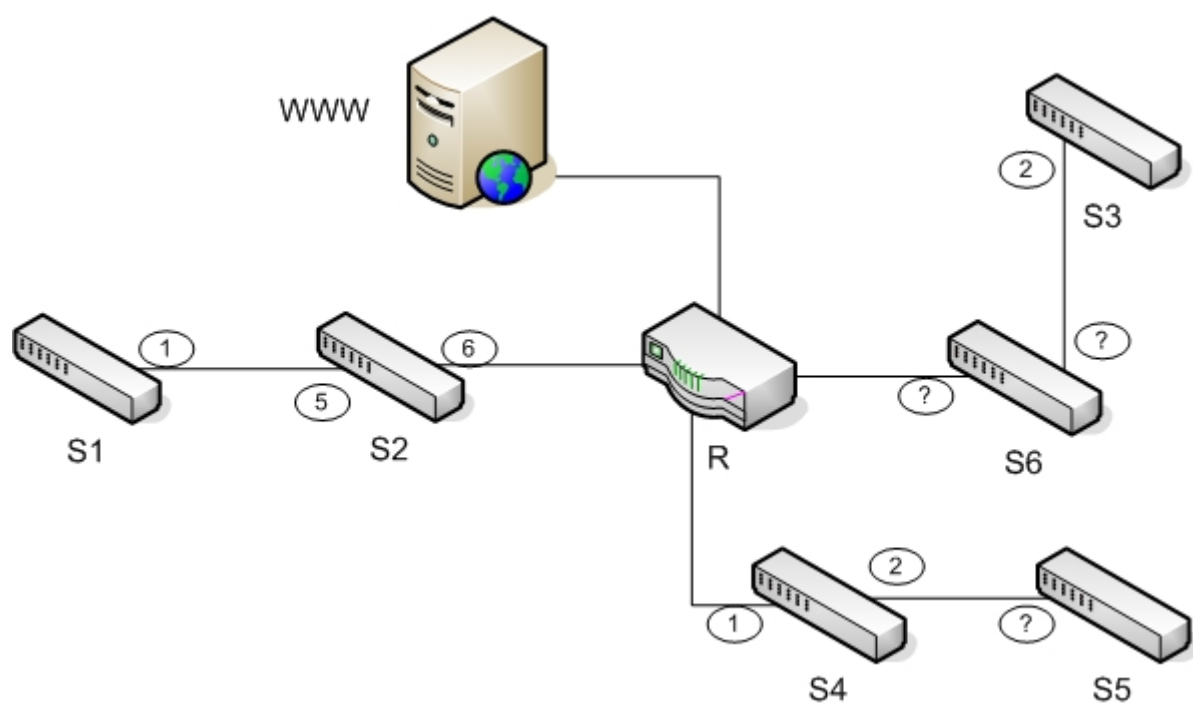


Figura 3.9 – Exemplo de uma rede em estrela com um *router* ao centro ligado à Internet.

4. Monitorização da Rede

4.1. Recolha de Informação

Para uma correcta utilização dos dados de uma rede, estes necessitam de estar completos e actualizados. Caso contrário, os resultados dessa utilização podem vir a revelar-se piores do que numa situação de inexistência de conhecimento, pois as conclusões baseadas em dados que já não corresponderem à realidade podem ser de todo erróneas. É necessária uma monitorização constante e atenta dos sistemas a gerir, para garantir o controlo e eficácia dos mesmos. Neste projecto, a recolha de informação é realizada por SNMP, permitindo acesso a um grande número de objectos e propriedades à frequência desejada.

Numa rede em funcionamento normal, a estrutura e topologia variam com pouca regularidade sendo, geralmente, apenas necessário efectuar uma nova descoberta uma vez por dia ou por semana. No entanto, se desejado, não deixa de ser possível realizá-la manualmente após qualquer mudança de configuração ou intervenção na rede, resultando numa anunciada alteração da topologia. O mesmo pode ser aplicado aos restantes dados da rede, tais como por exemplo, listas de equipamentos e respectivas descrições.

Todavia, existem parâmetros, fortemente variáveis na rede, que merecem uma atenção especial. O mais relevante corresponde ao tráfego entre máquinas que pode, por si só, ser responsável pela qualidade e funcionamento de uma rede. Torna-se assim essencial manter uma constante actualização das medidas dos equipamentos, para ser possível uma adequada utilização.

Tendo por objectivo garantir um fluxo regular de informação de tráfego, interroga-se a rede periodicamente por SNMP, com intervalos reduzidos, de apenas alguns minutos. O primeiro passo consiste em seleccionar qual a informação a recolher, de modo a tornar possível uma frequência elevada, sem sobrecarregar a rede. Sendo os equipamentos geridos essencialmente *switches*, opta-se por recolher da rede o tráfego de entrada e de saída de cada uma das suas interfaces. Consequentemente, torna-se possível identificar as zonas com maior carga na rede, assim como os equipamentos responsáveis por esta situação, não deixando de ter em conta a perspectiva global de todas as ligações.

A periodicidade entre as interrogações à rede deve ser definida tendo em consideração o tempo de demora de cada obtenção de informação. Não é possível recomençar a recolha de informação sem que a anterior tenha acabado, caso contrário, falhas de coerência e de funcionalidade podem ocorrer. Os resultados podem ser melhorados se todos os equipamentos forem acedidos simultaneamente, recorrendo para o efeito a diversos processos ou *threads*. Nesse caso, não é necessário esperar pela

chegada de uma informação para requisitar a seguinte, acelerando assim notavelmente o processo. Esta situação provoca um aumento na carga da rede, contudo, verifica-se que este incremento não é relevante numa rede como a do IST, por não ser comparável ao tráfego existente. Dependendo da utilização dos dados recolhidos, o intervalo de levantamento pode ser aumentado sem qualquer problema, mas esta opção implica obviamente a perda de resolução temporal na análise da rede.

4.2. Armazenamento dos Dados

4.2.1. Introdução

A recolha de informação é essencial na monitorização, no entanto, se os dados não forem correctamente guardados e indexados, dificilmente podem ser tratados e utilizados. Um dos objectivos deste trabalho corresponde à criação e manutenção de uma base de dados exportável, contendo informação sobre a rede e que possa ser expansível a outras ferramentas e aplicações de gestão.

Neste contexto, escolhe-se uma base de dados de *software* aberto (*open-source*), o MySQL, que possibilita um fácil preenchimento e acesso a partir de comandos em SQL (*Structured Query Language*). Sendo, actualmente, esta a linguagem por omissão na criação e consulta de bases de dados, foram criadas todo o tipo de bibliotecas, garantindo assim a sua correcta e eficiente utilização por diversas ferramentas e linguagens de programação. Como já referido, escolhe-se a biblioteca Mysqldb [Mysq06] para o Python, mas as operações criadas e realizadas, *queries*, podem ser executadas em qualquer outra plataforma suportando SQL.

A partir de um repositório de informação, podem efectuar-se facilmente procuras directas nos dados. No entanto, também é possível tratá-los e cruzá-los com outros tipos de aplicações para, no final, extrair novas informações e conclusões. Por conseguinte, esta aplicação justifica o uso de um formato de referência, visando potenciais compatibilidades com complementos e aplicações futuras.

4.2.2. Modelo de Dados

O estudo da rede baseia-se na informação recolhida dos *switches*, sendo estes naturalmente responsáveis por grande parte da ocupação na base de dados. Para cada um deles, é guardado o seu endereço ip, nome associado, descrição e tempo de resposta. Distinguem-se igualmente os seus diferentes portos, os quais se encontram ligados a outras interfaces de outros equipamentos.

É armazenado o tráfego de entrada e de saída de cada porto, tendo em conta a data e hora da medida, para posteriores operações. A topologia, por sua vez, é caracterizada pelas ligações hierárquicas entre um *switch-filho*, mais baixo hierarquicamente, e o seu *switch-pai* correspondente,

ascendente na direcção do centro da rede. Consequentemente, esta informação é suficiente para deduzir a infra-estrutura global da rede, não sendo necessário registar qualquer informação adicional sobre a topologia.

Como complemento, são obtidas e registadas diversas informações, com o objectivo de auxiliarem a realização de operações de gestão, as quais poderão ser comprovadas seguidamente. Nesse âmbito são armazenadas as diferentes VLANs existentes na rede, juntamente com o seu nome e localização. Estas são inter-ligadas aos portos dos *switches* respectivos, para permitir pesquisas rápidas sobre o seu alcance dentro da rede e para identificar as interfaces utilizadas para cada VLAN. É também registada a tabela de ARP do *router* central, para permitir a tradução de endereços MAC para endereços IP e vice-versa, requisito essencial para um correcto funcionamento da funcionalidade de localização de equipamentos na rede.

Da mesma forma, são registados os eventos relevantes relativos ao funcionamento da ferramenta. É escolhido um histórico de operações, relatando todas as escritas na base de dados, que tem como finalidade manter um registo de eventos e reconhecer a frescura dos dados existentes. Escolhe-se uma entrada comum para todas as ocorrências de um mesmo tipo, no intuito de se poder inquirir rapidamente essa informação, a partir de um único acesso à base de dados.

Uma das possibilidades de gestão consiste na análise do tráfego, na intenção de se detectarem anomalias, situação que irá ser estudada com mais detalhe posteriormente. Com o objectivo de inventariar estas ocorrências, são registadas as anomalias identificadas no tráfego, juntamente com o data e hora do sucedido, informação deveras útil em qualquer análise de incidentes na rede.

Actualmente, a capacidade de armazenamento é de tal ordem que não se justifica a implementação de mecanismos de limpeza automática, tais como os utilizados na ferramenta RRDTOOL, pois os dados deste projecto não são muito volumosos. Para a rede do IST, um disco de 100GB seria suficiente para um funcionamento contínuo durante mais de 10 anos; logo, não existe necessidade de perder resolução relativa a acontecimentos passados tendo, por único objectivo, uma redução do espaço utilizado.

4.2.3. Acesso à Informação

Como previamente referido, a informação pretendida da base de dados é adquirida graças a *queries* em SQL. As operações mais comuns correspondem ao acesso directo a uma linha de uma dada tabela, sendo a interpretação realizada pela ferramenta externa. Noutros casos, utiliza-se a base de dados para efectuar parte do processamento que, de outra forma, obrigaria a várias linhas de código e a diversas *queries*, aumentado naturalmente o tempo e complexidade necessários.

Para exemplificar o acesso, considera-se a *query* da Figura 4.1 que identifica o porto de saída de um *switch* se a informação sobre a topologia não se encontrar disponível. Como a rede do IST está configurada em estrela, verifica-se que, no porto de saída, o número de endereços na respectiva AFT é cerca de uma ordem de grandeza acima da quantidade de entradas nas AFTs de cada uma das restantes interfaces. Consequentemente, é apenas necessário localizar o porto com a maior AFT para obter o resultado desejado.

```
select numero
from ( select numero,count(mac_ligado) as n_macs_
      from interface_switch
      where ip = <ip_switch>
      group by numero
      having n_macs_ = (select max(n_macs)
                      from (select count(mac_ligado) as n_macs
                          from interface_switch
                          where ip = <ip_switch>
                          group by numero) as T)
      ) as T2
```

Figura 4.1 – *Query* à base de dados para obter o porto de saída de um *switch*.

O acesso à informação da base de dados proporciona, por si só, uma ajuda significativa à administração da rede. Torna-se possível, numa única pesquisa, listar os equipamentos existentes na rede, assim como as suas respectivas descrições. Do mesmo modo, possibilita-se por exemplo um conhecimento sobre a extensão das VLANs na rede, assim como os portos dos *switches* em que estão associadas.

No entanto, a informação armazenada pode ser utilizada para se efectuarem operações de gestão de complexidade e importância muito superiores. Destacam-se a localização de equipamentos na rede e a análise de tráfego, tendo por objectivo, nomeadamente, a identificação de anomalias. Todas estas funcionalidades irão ser abordadas com mais detalhe seguidamente.

4.3. Possibilidades de Gestão

4.3.1. Localização de Equipamentos

A descoberta da topologia, assim como a constante monitorização da rede, abrem um novo leque de potencialidades de gestão. Uma das funcionalidades oferecidas, como já referido, corresponde a localizar automaticamente um equipamento, somente a partir do seu endereço. Evita-se assim a consulta manual, *switch a switch*, até ao encontro da informação desejada, não deixando, mesmo nesse caso, de ser necessário algum conhecimento sobre a topologia para poder tirar conclusões.

A informação recolhida da rede é essencialmente de nível-2, sendo obtida directamente dos *switches* existentes. Dessa forma, a localização de um equipamento é baseada na pesquisa do respectivo endereço MAC, dada ser esta a informação disponível. Contudo, na gestão de redes é mais usual caracterizar e identificar os equipamentos a partir dos endereços ip (endereços lógicos, de nível-3) e não pelos endereços físicos das suas interfaces. Utiliza-se, nesse caso, a informação sobre as tabelas ARP do *router* central para transformar o endereço ip no endereço MAC correspondente. Este pode agora ser localizado rapidamente a partir do algoritmo desenvolvido, cujos pontos principais são apresentados em seguida.

Seja E o equipamento cuja localização é desconhecida. A pesquisa realizada consiste numa sucessão de operações que pode ser observada na Figura 4.2. Começa-se por procurar o conjunto M de *switches* que contêm E nas suas AFT, informação que se encontra na base de dados criada. Conclui-se imediatamente o processo se M for vazio (E não encontrado) ou se M apenas tiver um *switch* S (E ligado a S). Caso contrário, existem pelo menos dois *switches* potencialmente ligados ao equipamento. Usa-se a informação sobre os portos de saída e a topologia de rede para inferir, quando possível, a sua localização exacta.

Função `get_localizacao(MAC)`:

```

    query à base de dados para obter switches em que o MAC se encontra nas AFTs
    seja M o conjunto de switches obtido
    se M vazio:
        MAC não encontrado
        retorna
    se M tiver apenas 1 switch S:
        MAC foi localizado, retorna S

    query à base de dados para obter switches em que MAC se encontra em AFTs \
    correspondentes a portos diferentes do de saída
    seja L o conjunto de switches obtido
    se L vazio:
        procurar informação de topologia para identificar o switch S, em M, mais próximo \
        do centro da rede, que contenha todos os outros switches de M abaixo dele na \
        hierarquia
        se encontrado:
            foi localizado, retorna S
        senão:
            localização do MAC não pode ser identificada com detalhe
            retorna M
    se L tiver apenas 1 switch S:
        foi localizado, retorna S
    senão: // pelo menos 2 switches em L
        procurar informação de topologia para identificar o switch S, em L, mais baixo \
        hierarquicamente, que contenha todos os outros switches de L acima dele
        se encontrado:
            foi localizado, retorna S
        senão:
            localização do MAC não pode ser identificada com detalhe
            retorna L

```

Figura 4.2 – Localização de um equipamento na rede pelo endereço MAC.

Designa-se por L o conjunto de *switches* de M que não contêm E no seu porto de saída. Se em L apenas restar um equipamento, E foi localizado. Caso contrário, L contém vários *switches* e E apenas pode estar ligado ao mais baixo hierarquicamente. Se assim não fosse, o *switch* em L mais afastado do centro da rede teria E na sua AFT do porto de saída, não podendo então fazer parte de L.

No entanto, existe a possibilidade de L ser vazio, situação que ocorre quando E está ligado ao centro da rede, sendo este último apenas detectado nos portos de saída. Neste caso, todos os *switches* são potenciais candidatos a terem uma ligação directa com E. Mais uma vez, a topologia permite reduzir as possibilidades aos *switches* mais elevados na hierarquia, dado estes se encontrarem obrigatoriamente no caminho entre os *switches* mais afastados do centro e E.

Em todo este procedimento considerou-se que a informação sobre a topologia era conhecida. Se este pressuposto não for verificado, torna-se impossível concluir a localização exacta de E, podendo apenas retornar os potenciais *switches* a que E possa estar ligado. Esta informação não deixa de ser útil para acelerar o processo de localização visto que minimiza o número de equipamentos a pesquisar manualmente.

Para ilustrar as diferentes situações possíveis, escolhe-se o exemplo da Figura 3.9. Considerando a informação das AFTs, obtém-se: $M = \{S1, S2, S3, S4, S5, S6\}$.

- Se E ligado entre S1 e S2: L apenas pode conter S2 podendo assim concluir-se sobre a localização de E (ligado a S2 pelo porto 5).
- Se E ligado directamente a S2 pelo porto 3: L apenas pode conter S2 podendo assim concluir-se a localização de E (ligado a S2 pelo porto 3).
- Se E ligado a S1 pelo porto 2: L contém S1 e S2; como S1 e S2 se encontram ligados, sendo S1 o mais baixo na hierarquia, E só pode estar ligado a S1.
- Se E ligado directamente a S2 pelo porto de saída: L = vazio e $M = \{S1, S2, S3, S4, S5, S6\}$; não se pode concluir com detalhe sobre a localização mas, graças à informação sobre a topologia, pode concluir-se que E apenas pode estar ligado ao *router* R e a $\{S2, S4, S6\}$, reduzindo assim o número de possibilidades.

4.3.2. Análise de Tráfego

Uma rede é criada para permitir a troca e a partilha de dados, informação que circula entre os equipamentos pelas suas inter-ligações. Em consequência, os valores de tráfego representam, em boa aproximação, o estado e funcionamento de uma rede. Estes podem ser úteis, não só para criar estatísticas na rede, mas também para detectar anomalias ou falhas em sistemas.

A ferramenta desenvolvida efectua *polling* regular permitindo obter os valores dos contadores de tráfego, à entrada e saída de cada porto dos *switches*. Para esta informação ser utilizável, são necessárias no mínimo duas medidas, podendo assim retirar-se a diferença dos contadores,

correspondendo, naturalmente, ao tráfego realizado entre as duas recolhas de informação. É necessário ter em especial atenção que os contadores são geralmente de 32 bits. Como tal, se o valor medido for inferior ao anterior, deduz-se que o contador ultrapassou o seu valor máximo reiniciando de imediato. Nestes casos, basta adicionar 2^{32} ao valor da diferença entre contadores para se obter o valor correcto. A largura de banda média utilizada pode também ser calculada dividindo o tráfego de uma amostra pela duração da mesma.

A partir de toda esta informação, é possível realizar *data mining* e inferir diversas informações sobre o estado global da rede. As zonas de maior sobrecarga são identificáveis, facilitando, por exemplo, um estudo sobre a distribuição de equipamentos ou sobre uma possível expansão à rede. Esta informação auxilia fortemente o planeamento de acções de melhoria, devido a ser baseada em dados concretos e constantemente actualizados.

Uma outra aplicação consiste em identificar ou mesmo prever alterações significativas no tráfego, resultantes de anomalias tais como más configurações ou ataques à rede. Contudo, este processo é complexo e moroso, devido ao facto do tráfego ser fortemente irregular e dificilmente previsível. Numa tentativa de responder a esta situação, serão estudadas diversas abordagens ao problema, cujos resultados serão analisados em detalhe nos próximos capítulos.

5. Detecção de Anomalias

5.1. Introdução

Hoje em dia, as novas tecnologias de informação são responsáveis pela grande maioria da circulação de informação e por um grande número de serviços, imprescindíveis para o funcionamento de uma organização. As redes de comunicação correspondem ao pilar de suporte destes serviços, tornando assim vital garantir o seu funcionamento contínuo e adequado às necessidades.

Existem, contudo, diversos riscos associados a estas redes, devidos nomeadamente a erros de parametrização, avarias, ataques ou violações de segurança. Com o objectivo de mitigar estes riscos, é importante monitorizar continuamente o estado das redes assim como o seu funcionamento. Devido ao aumento da complexidade das redes, esta tarefa tem vindo a ser significativamente dificultada, requerendo cada vez mais recursos e ferramentas especializadas.

Existem dispositivos de segurança, tais como *Intrusion Detection System* (IDS) ou *Intrusion Prevention System* (IPS) que tentam detectar e bloquear não conformidades na rede, provocadas por ataques de *hackers*, vírus ou *worms* em circulação. No entanto, a detecção de intrusões é essencialmente baseada em assinaturas de ataques ou *worms* conhecidos, mantidas numa base de dados frequentemente actualizada, ou num conjunto de regras pré-definidas, reflectindo a actividade de uma aplicação maliciosa. Ora todos os dias são identificadas novas vulnerabilidades e desenvolvidos novos ataques, cuja detecção não é possível a partir de IDS/IPS, antes da respectiva assinatura ou esquematização ser reconhecida e integrada na base de dados da aplicação.

Por outro lado, as anomalias nos equipamentos de rede são, no geral, detectadas por ferramentas de monitorização que questionam periodicamente os dispositivos, disparando alarmes se o funcionamento não estiver dentro dos limites definidos. Porém, em sistemas complexos, ocorrem frequentemente situações em que todos os parâmetros se encontram dentro de valores regulares quando individualmente considerados, mas em que uma dada combinação de vários indicadores indicia situações irregulares de funcionamento. Estas podem derivar de avarias, aplicações maliciosas ou outros motivos operacionais, mas dificilmente são identificáveis por sistemas de segurança do tipo IDS/IPS. Este tipo de situações é por vezes detectável por um operador humano experimentado, mas nem sempre é fácil a sua tradução num conjunto determinístico de regras explícitas, dificultando a sua automatização.

Na intenção de auxiliar a resolução destas questões, pretende-se dotar a ferramenta desenvolvida de um mecanismo autónomo, permitindo detectar situações de funcionamento anómalo, informando os administradores de rede da ocorrência de tal irregularidade. O objectivo é auxiliar a gestão da rede,

detectando o mais depressa possível as anomalias, para poderem ser iniciados os processos de resolução das mesmas, minimizando a sua propagação e a interferência na qualidade de serviço.

5.2. Estado da Arte

5.2.1. Metodologias Desenvolvidas

Devido à relevância do tema, numerosos métodos e técnicas foram estudados e desenvolvidos com o objectivo de detectar anomalias numa rede. Uma primeira ideia simples consiste em comparar os valores de tráfego com os valores obtidos nas horas, dias, semanas e meses anteriores, a fim de detectar disparidades importantes. A comparação deve ser efectuada com diferentes amostras, para ter em conta as oscilações naturais da rede, entre períodos de pico, tais como as tardes dos dias de semana laboral, e os períodos de vazio, tais como as noites, os fins de semana ou as férias escolares.

Em [CoMu04], procuram-se assim *deltóides* na rede, correspondendo a alterações significativas no tráfego, sendo estas absolutas, relativas ou variacionais. O pretendido consiste em relacionar os *deltóides* com as modificações no funcionamento dito normal da rede, reflectindo assim uma potencial anomalia. Numa outra perspectiva, [KSZC03] centra-se na criação de resumos do andamento do tráfego para implementar uma série de modelos de previsão que, quando não verificados por um erro superior a um certo patamar, resultam no lançamento de um alarme. De uma forma comparável, [JiPa03] tenta calcular os valores relativos a um tráfego sem anomalias, filtrando as amostras obtidas da rede. Seguidamente, realiza uma comparação dinâmica entre o tráfego real e os valores resultantes do modelo criado, identificando assim as discrepâncias existentes.

Estes métodos focam-se essencialmente em troços de redes de *backbone* ou de elevada largura de banda, em que o tráfego resulta da confluência de diversas sub-redes. Consequentemente, nos resultados apresentados, o tráfego é sensivelmente periódico e regular, devido ao facto das pequenas anomalias serem desprezáveis face à largura de banda em questão. Deduz-se assim que estes métodos não são facilmente adaptáveis a pequenas e médias redes de natureza fortemente irregular, tal como no campos universitário do IST, em que o tráfego varia de uma forma totalmente imprevisível. Se for detectado um aumento de carga num equipamento, por exemplo, dificilmente é distinguível por estes métodos se esta situação se deve ao *download* do lançamento de uma nova versão de um programa popular ou, em oposição, se é originada por um *Denial of Service* (DOS) ou *worm* malicioso. Mais uma vez se reforça a necessidade de correlacionar outro tipo de informação sobre equipamentos e sistemas, antes de se poder tirar alguma conclusão relevante, tal como demonstra o estudo realizado em [Maxi90].

Uma abordagem diferente foi seguida e apresentada em [GHYC06] e [LLLi07]. Nestes projectos, o tráfego é decomposto em diversas componentes de frequências variadas, de uma forma semelhante à transformada de Fourier. Estas componentes correspondem a diversas resoluções do problema, cada uma permitindo detectar anomalias de um certo tipo. Embora de implementação complexa, ambos os artigos apresentam resultados satisfatórios para ataques ou sobrecargas na rede. No entanto, o tráfego é analisado isoladamente por troço de rede, limitando assim a utilização dos algoritmos à detecção de anomalias locais. Com esta metodologia, não seriam detectados os casos em que os valores do tráfego de diversos equipamentos se encontrariam dentro dos padrões normais de funcionamento, se observados independentemente, mas cujo estado global da rede estaria afectado, o que poderia ser identificável correlacionando toda a informação disponível.

Na tentativa de responder ao problema levantado, estuda-se a possibilidade de utilizar sistemas de aprendizagem automática. O objectivo passa pelo desenvolvimento de mecanismos de reconhecimento de padrões no tráfego, de forma a possibilitar a criação de um modelo que caracterize o funcionamento “normal” de uma rede. Os resultados deste tipo de abordagem são menos determinísticos, mas possibilitam uma detecção mais vasta, devido a originarem de um processo de aprendizagem e não do resultado de um conjunto de regras explícitas. Esta aprendizagem pode ser realizada por métodos e técnicas diversos, incluindo redes neuronais ou sistemas de inteligência artificial [Bish95].

5.2.2. Fundamento Utilizado

Na intenção de desenvolver uma detecção de anomalias o mais generalista possível, escolhe-se realizar um estudo sobre os sistemas de aprendizagem automática. Neste caso, não são procuradas as anomalias de forma directa, tais como os picos ou falhas no tráfego de um equipamento. A abordagem é realizada de forma global, em que é elaborado um modelo do tráfego dito normal na rede e é estudada a sua evolução ao longo do tempo. Seguidamente, possibilita-se a identificação dos comportamentos da rede que se afastem do modelo desenvolvido, indiciando situações anómalas de funcionamento.

Neste tipo de sistemas, nem sempre se torna viável obter um conjunto de dados anómalos reais para treinar o modelo. Primeiro, porque se pretende detectar anomalias desconhecidas ou não facilmente exemplificáveis, segundo, porque não é praticável denegrir o funcionamento da rede para produzir os dados procurados. Assim sendo, torna-se necessário realizar uma aprendizagem não supervisionada, em que se procuram os padrões dos dados de entrada e não a adaptação do modelo às anomalias conhecidas.

Consideram-se variáveis aleatórias os valores do tráfego em diversos troços da rede. Pretende-se estudar a possibilidade de estimar uma função de densidade de probabilidade capaz de caracterizar o funcionamento normal do sistema, amostrado por estas variáveis. Como função de densidade de

probabilidade, escolhe-se uma mistura de gaussianas a d dimensões, pois esta pode adaptar-se aos padrões nos dados, variando os parâmetros da mistura consoante a necessidade.

Vão agora apresentar-se os passos principais da metodologia desenvolvida neste projecto. Começa-se por escolher os parâmetros de entradas do modelo para, em seguida, procurar a mistura de gaussianas que mais se aproxima destes valores. Finalmente, resta calcular a função de verosimilhança resultante, de forma a poder designar limiares abaixo dos quais as situações sejam consideradas anomalias no funcionamento, devido aos dados se afastarem do modelo criado.

5.3. Modelo de Tráfego

5.3.1. Parâmetros de Entrada

Idealmente, poderia considerar-se a hipótese de escolher os valores de tráfego de todos os portos dos *switches* existentes, como entradas para o modelo. No entanto, se fosse esse o caso, obter-se-ia um modelo com uma dimensão da ordem dos milhares, totalmente inviável na prática. Não só seria necessário um processamento de dados muito intenso, com uma base de dados crescendo rapidamente, mas também longos anos de utilização, para conseguir dados de treino suficientes para a modelação de um sistema de tal dimensão. Isto deve-se particularmente ao facto do número de amostras necessárias para obter os parâmetros de um modelo aumentar exponencialmente com o número de dimensões.

Como alternativa, escolhe-se um conjunto de pontos na rede, localizados nos troços principais onde veicula a informação. Estas ligações de rede são responsáveis por grande parte do volume de tráfego, reflectindo assim o estado global do sistema. Dessa forma, reduz-se significativamente as dimensões do modelo, possibilitando a sua aplicação prática.

Por outro lado, sendo este projecto focado numa rede universitária, é importante ter em conta as oscilações temporais na rede. Mesmo sendo imprevisível, o tráfego não deixa de ter comportamentos diferentes consoante a data e hora das amostras. É de esperar que cerca das 05h00 de domingo, o tráfego possa ser significativamente inferior ao mesmo levantado pelas 16h00 de um dia de semana, onde a maioria dos estudantes se encontra presente no campus universitário.

Verifica-se, na prática, que os resultados obtidos em modelos de tráfego compostos por misturas de gaussianas são mais exactos se contemplarem a data e hora das amostras, pois as gaussianas conseguem moldar-se mais eficientemente aos dados. Com o objectivo de aumentar a precisão do modelo, adiciona-se então como entradas ao modelo, a hora e dia da semana da amostra, correspondendo às diferenças mais significativa no tráfego de uma rede.

5.3.2. Algoritmo Expectation-Maximization

Como indicado anteriormente, o objectivo deste estudo consiste em procurar a mistura de gaussianas que melhor caracterize os dados de entrada. Utilizando a notação de [Bish95], a função de densidade de probabilidade de uma gaussiana a d dimensões pode ser escrita por

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right\} \quad (5.1)$$

em que $\boldsymbol{\mu}$ corresponde à média da gaussiana (a d dimensões) e Σ à respectiva matriz de covariância. Seguindo a mesma notação, a função de densidade de probabilidade de uma mistura de M gaussianas pode ser escrita por

$$p(\mathbf{x}) = \sum_{j=1}^M p(\mathbf{x} | j) P(j) \quad (5.2)$$

onde $p(\mathbf{x} | j)$ corresponde à função de densidade de probabilidade da gaussiana j (5.1) e $P(j)$ à probabilidade à priori da gaussiana j relativamente às outras gaussianas da mistura, com

$$\sum_{j=1}^M P(j) = 1 \quad (5.3)$$

A dificuldade nesta fase reside na determinação dos parâmetros da mistura de gaussianas (vectores média, matrizes de covariância e probabilidades associadas), de forma a se adaptarem o mais possível aos dados. Existem variados métodos para calcular estes mesmos parâmetros; o algoritmo Expectation-Maximization (EM) é um dos mais utilizados neste tipo de aplicações, por ser determinista e iterativo, o que o torna facilmente convertível numa linguagem de programação.

Seja N o número de amostras de dados disponíveis; forma-se assim o seguinte conjunto de dados de entrada

$$\Upsilon = \{ \mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \dots, \mathbf{x}^N \} \quad (5.4)$$

O algoritmo EM [Bish95] começa por iniciar o conjunto de parâmetros com valores arbitrários, levando à estimativa inicial da mistura $p^0(\mathbf{x})$. Seguidamente, calculam-se as respectivas probabilidades à posteriori utilizando o teorema de Bayes, obtendo-se

$$P^t(j | \mathbf{x}^n) = \frac{P^t(j) p_j^t(\mathbf{x}^n)}{p^t(\mathbf{x}^n)} \quad (5.5)$$

em que $p_j(\mathbf{x}^n)$ é a função de densidade de probabilidade da gaussiana j no ponto n , enquanto $p(\mathbf{x}^n)$ é a função de densidade de probabilidade da mistura no mesmo ponto. Por último, t corresponde à iteração do algoritmo.

A partir da maximização das verosimilhanças, torna-se possível estimar os parâmetros restantes. A demonstração de todo o processo pode ser encontrada em [Bish95], sendo o resultado final descrito pelas expressões que se seguem:

$$\boldsymbol{\mu}_j^{t+1} = \frac{\sum_{n=1}^N P^t(j | \mathbf{x}^n) \mathbf{x}^n}{\sum_{n=1}^N P^t(j | \mathbf{x}^n)} \quad (5.6)$$

$$\boldsymbol{\Sigma}_j^{t+1} = \frac{\sum_{n=1}^N P^t(j | \mathbf{x}^n) (\mathbf{x}^n - \boldsymbol{\mu}_j^t)(\mathbf{x}^n - \boldsymbol{\mu}_j^t)^T}{\sum_{n=1}^N P^t(j | \mathbf{x}^n)} \quad (5.7)$$

$$P(j)^{t+1} = \frac{1}{N} \sum_{n=1}^N P^t(j | \mathbf{x}^n) \quad (5.8)$$

Pode assim concluir-se que o algoritmo EM permite a procura dos parâmetros das gaussianas, iterando as expressões (5.5) a (5.8) até um dado critério de paragem. Este método possibilita então a criação de um modelo composto por uma mistura de gaussianas, identificando os principais padrões nos dados amostrados.

5.3.3. Minimum Message Length

Embora o algoritmo EM realize a procura dos parâmetros de uma mistura de gaussianas, é sempre necessário escolher manualmente o número de gaussianas do modelo a determinar. Nos testes realizados, verificou-se que, para obter a mistura que melhor representava o conjunto de dados (maior verosimilhança), era necessário escolher um número de gaussianas diferente de modelo para modelo, consoante os padrões existentes nos dados. Torna-se assim difícil automatizar o processo sem ter de escolher, arbitrariamente, um valor pré-definido de gaussianas.

Na intenção de ultrapassar esta questão, propõe-se complementar o algoritmo apresentado com um método de comparação de modelos, de forma a poder identificar qual o que contém o número de gaussianas mais adequado. Numa primeira fase, é importante definir um factor de avaliação.

Baseando-se na Teoria da Informação de Shannon [ShWe49], a informação pode ser definida por

$$\text{informação} = -\log(\text{probabilidade}) \quad (5.9)$$

Esta teoria indica que a informação pode ser codificada, de forma a ser representada por uma quantidade menor de dados, realizando assim uma compressão. Num código ideal, o comprimento da informação (em bits) de um modelo θ , com probabilidade $P(\theta)$, pode então ser descrito por

$$\text{comprimento}(\theta) = -\log_2(P(\theta)) \quad (5.10)$$

Seguindo o princípio da Navalha de Occam, a representação de um modelo deve utilizar o menor número de premissas possível, dando primazia, num conjunto de modelos equivalentes, ao que se revelar mais simples. Espera-se assim minimizar o comprimento do modelo, mantendo uma correcta adequação aos dados.

Em resposta a esta demanda, Wallace desenvolveu uma metodologia para comparação de modelos, o Minimum Message Length (MML) [Wall05]. Se se considerar uma troca de informação entre duas identidades, o objectivo apresentado consiste em minimizar o comprimento dos dados enviados pelo emissor, sem que estes deixem de ser correctamente interpretados pelo receptor. De forma a comprimir essa informação, realiza-se a codificação dos dados numa mensagem que se pode decompor em duas componentes: um modelo, tentando representar os dados tanto quanto possível, e um conjunto adicional de informação, pormenorizando a diferença entre os dados reais e a aproximação obtida pelo modelo.

Seja E uma mensagem de dados que se pretende codificar utilizando o modelo θ . Esta mensagem pode limitar-se a conter a informação sobre o modelo θ , juntamente com a informação necessária para caracterizar a diferença D entre os dados de E e os resultados do modelo θ . Podemos definir o comprimento de E como

$$\text{comprimento}(E) = \text{comprimento}(\theta) + \text{comprimento}(D | \theta) \quad (5.11)$$

Em (5.11), $\text{comprimento}(\theta)$ pode ser interpretado como a complexidade do modelo θ . Quanto maior for o $\text{comprimento}(\theta)$, maior será o número de parâmetros do modelo, resultando numa

maior precisão. De forma análoga, o $\text{comprimento}(D|\theta)$ pode ser considerado como o erro do modelo, correspondendo à informação necessária para descrever os dados, dado o modelo θ . Se o modelo θ for simples e estiver pouco adaptado aos dados, o $\text{comprimento}(\theta)$ será reduzido, mas o erro terá uma complexidade elevada, aumentando, conseqüentemente, o comprimento desta componente da mensagem.

Conclui-se assim que, para minimizar o comprimento das mensagens, é necessário balancear a complexidade do modelo relativamente à sua adaptação aos dados. Este método permite identificar o modelo preferencial entre um conjunto de modelos equivalentes, realizando a comparação a partir do comprimento das mensagens.

O $\text{comprimento}(E)$ pode ser reformulado utilizando (5.10) para se obter

$$\text{comprimento}(E) = -\log_2(P(\theta)) - \log_2(P(D|\theta)) \quad (5.12)$$

Tanto a probabilidade do modelo ($P(\theta)$) como a probabilidade do erro dado o modelo ($P(D|\theta)$) podem ser modeladas por diversas funções ou distribuições. Pode considerar-se, por exemplo, um sistema contínuo em que as funções de densidade de probabilidade $p(\theta)$ e $p(D|\theta)$ são compostas por misturas de gaussianas. Nesta situação, quanto mais complexo é o modelo, maior é o número de gaussianas utilizadas. Este parâmetro resulta, pela fórmula (5.2), numa função de densidade de probabilidade $p(\theta)$ com valores inferiores, o que aumenta naturalmente o comprimento do modelo definido por (5.10). Por outro lado, se o modelo for complexo e se encontrar fortemente adaptado aos dados, a mistura de gaussianas que modela o erro poderá ser simples, composta por apenas uma ou duas gaussianas, por exemplo. Esta situação resulta, obviamente, numa função de densidade de probabilidade $p(D|\theta)$ superior, diminuindo, conseqüentemente, o comprimento desta componente.

Uma ideia simples de integração no algoritmo EM consiste em escolher um número elevado de gaussianas (número este que pode ser relacionado com o número de dados) e em seguida iniciar o método EM com algumas alterações. A cada iteração do algoritmo EM, realizar, por exemplo, o seguinte conjunto de operações:

- Calcular o comprimento de uma mensagem contendo todos os dados, para servir de referência: $\text{comprimento}_{ref}(E)$. Esta mensagem deve ser codificada pelo modelo composto pelos parâmetros resultantes da respectiva iteração do algoritmo EM.
- Recalcular o valor do comprimento, mas agora excluindo uma gaussiana da mistura. Se o comprimento global for inferior ou igual ao valor de referência

($\text{comprimento}(E) \leq \text{comprimento}_{ref}(E)$), ou seja se a redução da complexidade do modelo ($\text{comprimento}(\theta)$) for mais relevante que o aumento no erro resultante ($\text{comprimento}(D|\theta)$), retira-se a gaussiana que foi excluída do modelo. Em seguida, repete-se este passo para todas as gaussianas até que seja identificado a mistura que resulte no comprimento mínimo da mensagem. Possibilita-se assim a redução do número de gaussianas no modelo, mantendo uma importante adequação aos dados.

- De uma forma semelhante, o processo anterior pode ainda ser otimizado. Escolher, neste caso, um par de gaussianas da mistura e substituí-las por uma gaussiana que abranja os dados das duas gaussianas anteriores, dada uma certa precisão. Mais uma vez, se o comprimento global for inferior, altera-se a mistura para contemplar a nova gaussiana, retirando o par excluído. A precisão não é o mais relevante nesta fase, pois trata-se de uma redução de complexidade e não da convergência para o valor final. Tal como no passo anterior, repete-se este processo até que todos os pares tenham sido testados.

Este método permite que, numa primeira fase, as iterações sejam essencialmente dedicadas à procura do número mais adequado de gaussianas. Seguidamente, é realizada a convergência dos parâmetros das gaussianas para os valores finais, utilizando, para o efeito, as potencialidades do algoritmo EM. Obtém-se assim a estimativa de máxima verosimilhança que minimiza o $\text{comprimento}(E)$, respondendo dessa forma ao problema levantado.

Para a integração deste processo na ferramenta desenvolvida, escolheu utilizar-se a biblioteca para Python PyMML [Harr05]. Este módulo implementa uma variante do método apresentado, resultando na identificação dos parâmetros de uma mistura de gaussianas adaptada aos dados, baseando-se na metodologia MML.

5.3.4. Identificação de Anomalias

A partir do método anterior, obtém-se o conjunto de parâmetros de uma mistura de gaussianas que representa, tanto quanto possível, o tráfego da rede. Esta mistura tem uma função de densidade de probabilidade que resulta na soma de diversas componentes (gaussianas), correspondendo aos agrupamentos principais de dados no tráfego amostrado.

Assim sendo, podemos concluir que os máximos da função de densidade de probabilidade correspondem aos valores “normais” de tráfego, devido a terem sido identificadas diversas amostras com valores nessa gama. Em oposição, os valores mais baixos dessa mesma função indicam valores pouco usuais no tráfego, afastados das observações que serviram de base ao modelo.

A partir desta correspondência, torna-se possível definir um limiar abaixo do qual o tráfego pouco usual seja considerado anômalo, indiciando uma situação irregular na rede. A escolha deste limiar é delicada, devido à importância de balancear o número de falsos positivos com o de falsos negativos. Os falsos positivos correspondem aos valores que se encontram abaixo do limiar, considerados, por essa razão, anomalias relativamente ao modelo, mas que equivalem na realidade ao funcionamento normal do sistema. Inversamente, os falsos negativos correspondem aos valores acima do limiar, não detectados como irregulares pelo modelo, que são, na realidade, o resultado de uma anomalia na rede.

Facilmente se percebe que, se o limiar for baixo, o número de falsos positivos será reduzido, enquanto o número de falsos negativos poderá ser importante. No caso contrário, verifica-se logicamente a situação inversa. Porém, sendo uma ferramenta de segurança utilizada para administração de redes, é indispensável que os falsos positivos não sejam demasiado importantes, mesmo que não sejam detectadas algumas das situações anormais de funcionamento. A razão desta preferência deve-se ao facto de, na existência de falsos positivos frequentes, as anomalias assinaladas começarem a ser ignoradas. Infelizmente, esta é uma situação recorrente, visível, por exemplo, quando não é dada a devida importância aos alarmes das firewalls e IDSs, contrariando assim o principal propósito de uma funcionalidade deste tipo.

5.4. Aplicações à Gestão de Redes

Neste capítulo, apresentou-se uma metodologia para proporcionar a criação de um modelo do tráfego, caracterizando o funcionamento normal do sistema. A aplicação de um limiar aos resultados obtidos permite a identificação de anomalias na rede, de forma automática, sem necessitar da intervenção de um operador humano.

Possibilita-se assim uma gestão mais eficiente da rede, pois os administradores são avisados atempadamente de uma situação irregular de funcionamento, podendo iniciar diligências para a sua resolução, antes desta provocar consequências mais prejudiciais. Como explicado anteriormente, o principal benefício desta ferramenta consiste na potencial detecção de todo o tipo de anomalias, mesmo desconhecidas, contemplando falhas operacionais na rede, aplicações maliciosas ou ataques de segurança.

Depois de identificada uma situação anómala no tráfego, os administradores de rede podem realizar uma pesquisa sobre a causa do sucedido, auxiliando-se igualmente da ferramenta desenvolvida. Se, por exemplo, se verificar uma anomalia resultando numa degradação grave da qualidade de serviço da rede, devem ser tomadas medidas o mais depressa possível, sob pena de impossibilitar o funcionamento de diversas aplicações. No caso do volume de carga anormal originar de um endereço específico, por exemplo, consegue-se, graças à funcionalidade de localização, identificar o *switch* ao

qual o equipamento em causa se encontra ligado. Pode então actuar-se na rede, bloqueando, temporariamente, o porto correspondente à ligação identificada, restabelecendo o funcionamento global numa base provisória, excepto na sub-rede em que se encontra o equipamento malicioso. A situação encontra-se agora confinada, sendo claramente mais vantajosa do que uma degradação generalizada da rede. Finalmente, resta aos administradores descobrir e eliminar a causa do problema, possivelmente uma aplicação maliciosa neste exemplo, para restaurar a trânsito total de informação e assim retomar a actividade normal da rede.

Por se tratar de um projecto académico, os alarmes sobre as anomalias detectadas são apresentados na consola de administração da ferramenta. No entanto, para um aviso mais eficiente, os alarmes podem ser enviados aos administradores de sistemas por correio electrónico ou por mensagem escrita para o telemóvel, de forma a possibilitar uma intervenção rápida, minimizando os efeitos nefastos da anomalia. Embora tenham sido desenvolvidas as funcionalidades essenciais neste projecto, a integração desta ferramenta na gestão e administração corrente da rede do IST terá de ser realizada juntamente com os administradores de sistemas do CIIST, de forma a ser útil e eficaz.

6. Apresentação dos Resultados

6.1. Interface do Utilizador

6.1.1. Página Web

Parte significativa deste projecto consiste na criação de uma ferramenta de monitorização, de descoberta de topologia e de detecção de anomalias. Para um utilizador ter acesso à informação disponibilizada, escolhe-se uma interface Web, devido à sua flexibilidade e acessibilidade. Não só é possível consultar e efectuar operações de qualquer computador ligado à Internet, mas também se torna facilmente alterável e moldável consoante as necessidades. Para o efeito, configura-se um servidor Apache suportando PHP e SSL (*Secure Sockets Layer*), correspondendo, no presente, a um dos servidores Web mais utilizados e testados.

O PHP é utilizado para facilitar a interacção com o utilizador, permitindo executar *scripts* do sistema operativo e apresentar os resultados de uma forma clara na página Web. No entanto, o PHP apenas conclui o carregamento da página quando todos os seus itens estão disponíveis, ou seja, quando todos os *scripts* executados retornam, bloqueando assim qualquer outro tipo de operação. Para resolver o problema, implementou-se um pequeno programa em C cuja única finalidade consiste na criação de um processo filho, o qual irá executar o script desejado, enquanto o processo principal retorna de imediato. O estado do script a decorrer é escrito e actualizado em ficheiro, permitindo a consulta por parte da interface do utilizador. Todas as funcionalidades, como efectuar uma nova descoberta, iniciar *polling* à rede ou consultar as anomalias detectadas, podem ser efectuadas em simultâneo, transformando a interface criada numa consola de gestão de toda a ferramenta.

Contudo, o facto destas funcionalidades estarem acessíveis na Internet constitui uma grave falha de segurança. A informação pode ser utilizada na intenção de obter um conhecimento sobre a estrutura da rede, para o planeamento de um ataque, ou no intuito de degradar a qualidade da rede, provocando, por exemplo, um excesso de tráfego nas ligações entre *switches*. A solução implementada consiste no uso de ligações seguras (SSL), para confidencialidade, e de uma sistema de autenticação dos utilizadores, para restringir o acesso às pessoas autorizadas.

Sendo indispensável a comunicação entre os *scripts* Python e a interface Web, definem-se dois ficheiros de configuração contendo todos os parâmetros variáveis, assim como a interface para troca de informação, os quais podem ser observados no Anexo A. Desse modo, qualquer modificação numa das partes não implica necessariamente uma alteração do código na outra, permitindo assim rápidas mudanças de parâmetros e expansões, sem passar por uma revisão global da aplicação.

A consola de gestão desenvolvida pode ser observada na Figura 6.1. Todas as funcionalidades apresentadas ao longo deste projecto estão presentes, juntamente com os respectivos resultados.



Figura 6.1 – Consola de gestão – janela relativa ao estado do *polling* à rede.

Toda a interface do utilizador foi desenvolvida de forma a ser intuitiva e auto-explicativa. No entanto, os detalhes do seu funcionamento, assim como uma listagem exaustiva das possibilidades oferecidas, podem ser consultados no manual que se encontra no Anexo B.

6.1.2. Representação da Topologia

A lista de equipamentos e ligações da rede é, no geral, extensa e de difícil leitura. Na intenção de facilitar esta tarefa e de modo a permitir uma mais rápida e eficiente identificação da topologia, desenha-se uma imagem global da rede, contento toda a informação pretendida.

Escolheu-se a ferramenta Prefuse [Heer06] que consiste num conjunto de bibliotecas usadas na criação de aplicações de visualização interactiva de informação. É inteiramente feita em Java, sendo, desse modo, facilmente integrável na interface Web sob a forma de um Applet.

Foram criados ficheiros XML (*eXtensible Markup Language*), contendo a informação obtida a partir da descoberta da topologia, para servirem de interface com a ferramenta Prefuse. Duas possibilidades de imagem de rede foram implementadas: uma primeira representando os *switches* pelos endereços IP para uma observação da informação de rede e uma segunda, equivalente, mas onde os endereços são substituídos pelos nomes dos equipamentos obtidos por SNMP, para uma representação mais intuitiva. A apresentação pode ser radial, mostrando claramente os níveis hierárquicos desde o centro da rede até à periferia, mas também dinâmica, visualmente mais atraente e intuitiva, mostrando

facilmente as ligações ponto a ponto existentes. Todas estas funcionalidades são observáveis no capítulo seguinte, em que são apresentados os resultados provenientes do algoritmo de descoberta de topologia desenvolvido.

6.1.3. Apresentação das Anomalias

Os resultados da detecção de tráfego anómalo na rede, indiciando falhas ou ataques à rede, podem ser observados na interface Web. São listadas as anomalias detectadas, juntamente com a data e hora de cada ocorrência. Um estudo mais aprofundado destes resultados será apresentado no próximo capítulo, onde será visível o retorno desta ferramenta.

Seguidamente, pode igualmente realizar-se uma pesquisa mais detalhada sobre o estado da rede no período detectado. Para permitir esta tarefa, diferentes informações encontram-se disponíveis para auxiliar na identificação da causa e dos efeitos resultantes da anomalia. Pode observar-se, por exemplo, a evolução do tráfego num dado porto, num intervalo de tempo específico (Figura 6.2). Para esta representação gráfica, escolhe-se a biblioteca JpGraph [Jpgr07] devido à sua fácil integração numa página PHP, não deixando de apresentar diversas possibilidades de configuração.

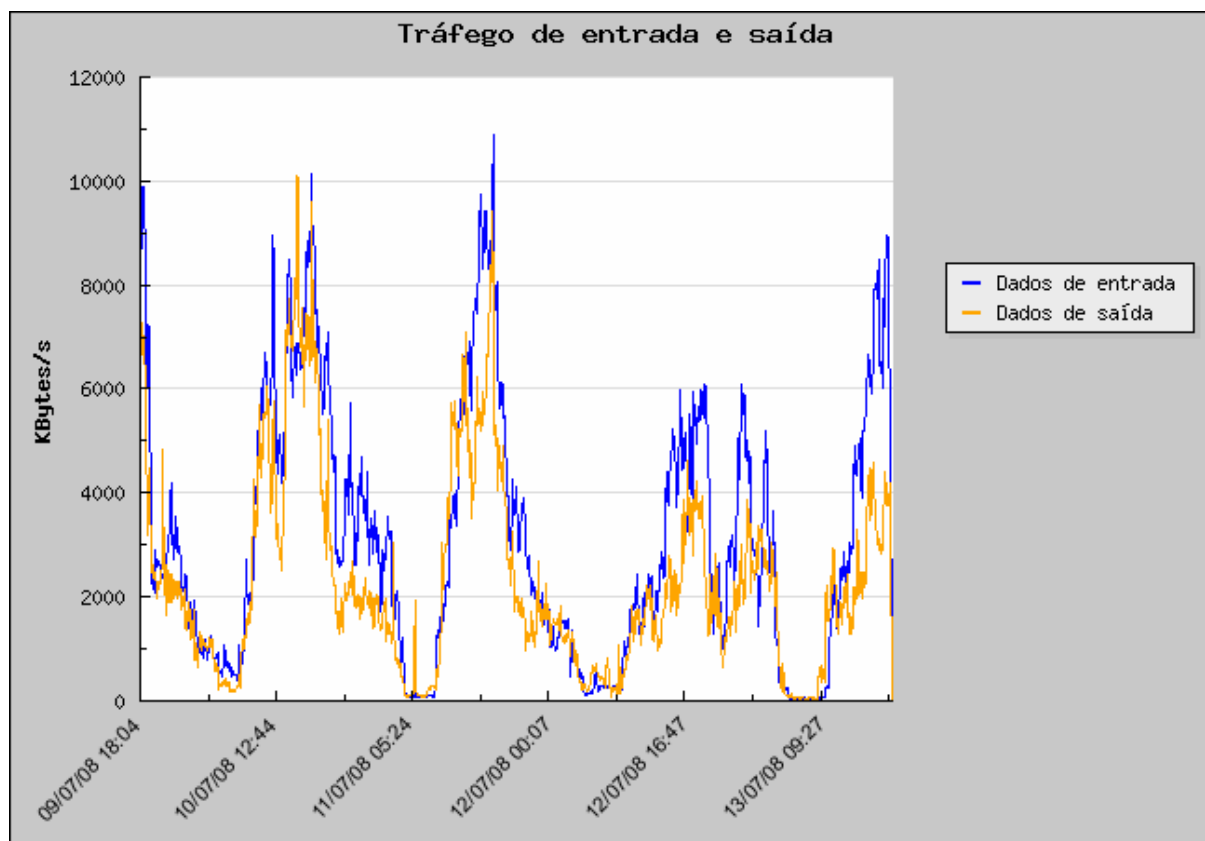


Figura 6.2 – Evolução do tráfego entre 09/07/08 18:00 e 13/07/08 18:00, na interface 2010 do *router* central do IST.

Nestes testes, verificou-se que, para uma rede moderadamente carregada e sem o auxílio de *spoofing*, nenhuma ligação directa foi encontrada a partir do Lema 3.1. Isto significa que todas as AFTs estavam incompletas e, como tal, não foi possível inferir directamente as ligações existentes sem recurso a heurísticas. No entanto, a pesquisa e organização de caminhos, assim como a execução das heurísticas, permitiram descobrir com sucesso a topologia da rede do IST, informação verificada e confirmada pelos administradores de sistemas do CIIST.

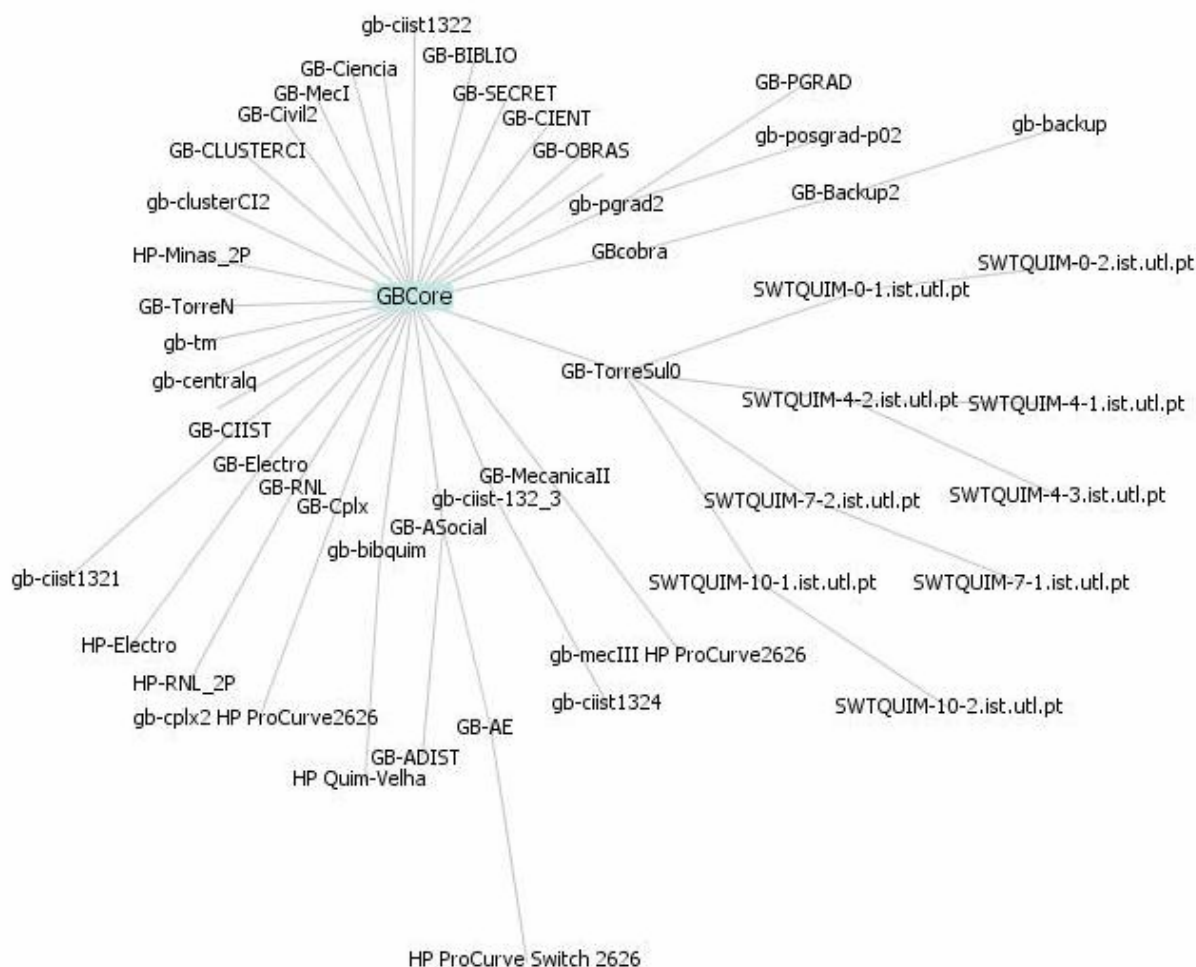


Figura 6.4 – Imagem da topologia da rede, em Janeiro de 2008, representando os equipamentos pelos respectivos nomes configurados.

Como explicado anteriormente, para uma rede em período de vazio, as AFTs encontram-se, em grande parte, desprovidas dos endereços necessários ao correcto desenrolar do algoritmo. Para se obterem resultados nestas condições, foi necessário recorrer ao *spoofing* entre equipamentos. Esta operação possibilitou a descoberta de algumas ligações, graças a AFTs completas, e a sua totalidade, utilizando o algoritmo completo, confirmando mais uma vez os resultados apresentados.

No desenrolar deste trabalho, foram adicionados diversos equipamentos à rede, expandindo a infraestrutura de *switches* existente no IST. A ferramenta desenvolvida foi acompanhando estas alterações, resultando na topologia que se pode observar na Figura 6.5 e na Figura 6.6, correspondente à rede do IST nos finais de Junho de 2008. Nestas condições, o algoritmo identificou os sessenta e três *switches* do IST, juntamente com as respectivas ligações.

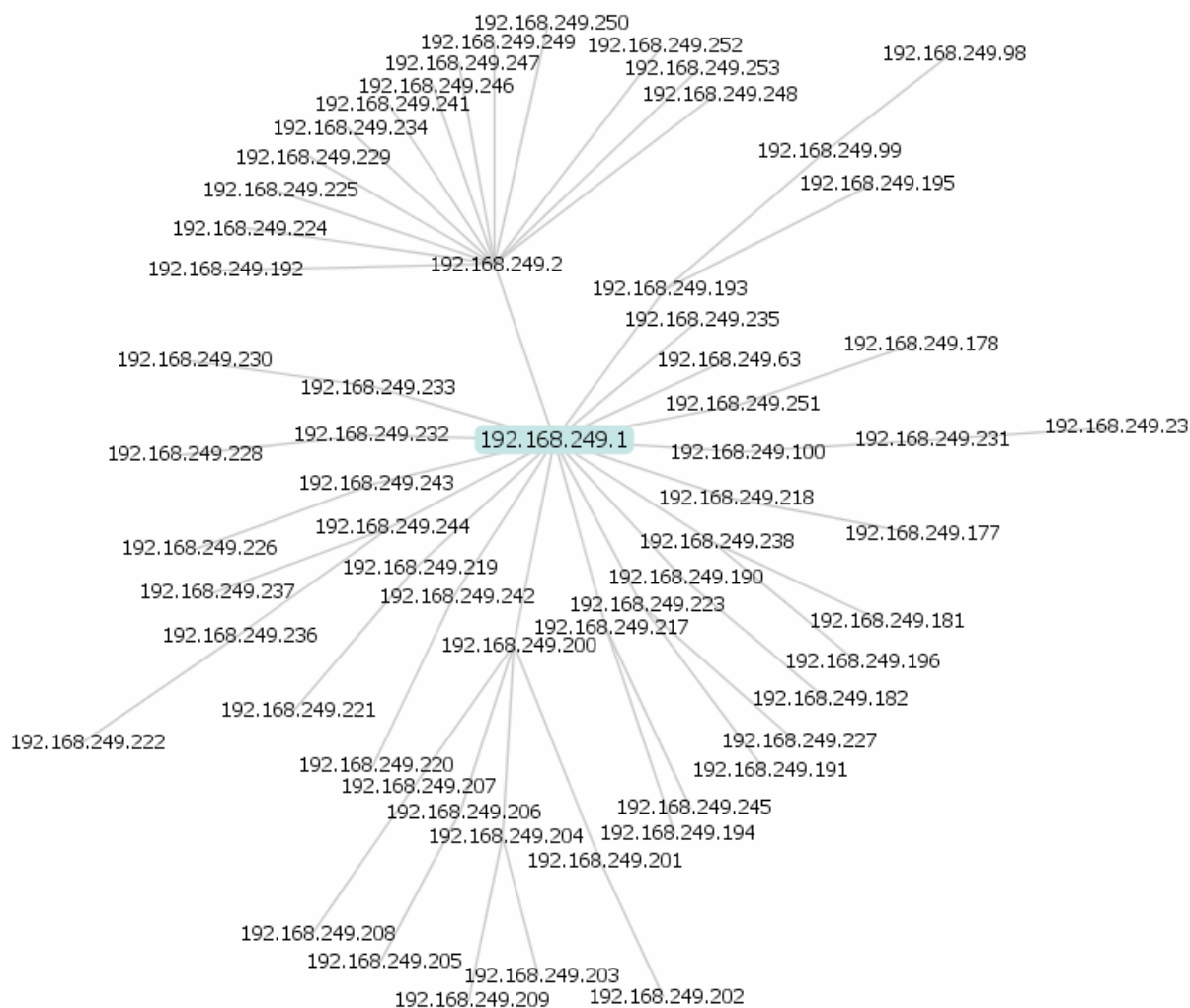


Figura 6.5 – Imagem da nova topologia da rede, em Junho de 2008, representando os equipamentos pelos respectivos endereços IP.

Mais uma vez se realça o facto do algoritmo ter detectado automaticamente esta mudança de topologia, ilustrando assim a sua utilidade prática. Valida-se assim que esta ferramenta, quando configurada para realizar uma descoberta de topologia regularmente, torna possível obter um conhecimento contínuo e actualizado sobre as ligações na rede e assim facilitar diversas operações de gestão.

Conclui-se assim que o algoritmo permite a identificação da topologia da rede do IST, acompanhando as alterações realizadas na infra-estrutura, desde que todos os métodos e heurísticas sejam utilizados. A partir dessa topologia, são possibilitadas diversas funcionalidades, tais como, por exemplo, a localização de equipamentos, conforme irá ser demonstrado seguidamente.

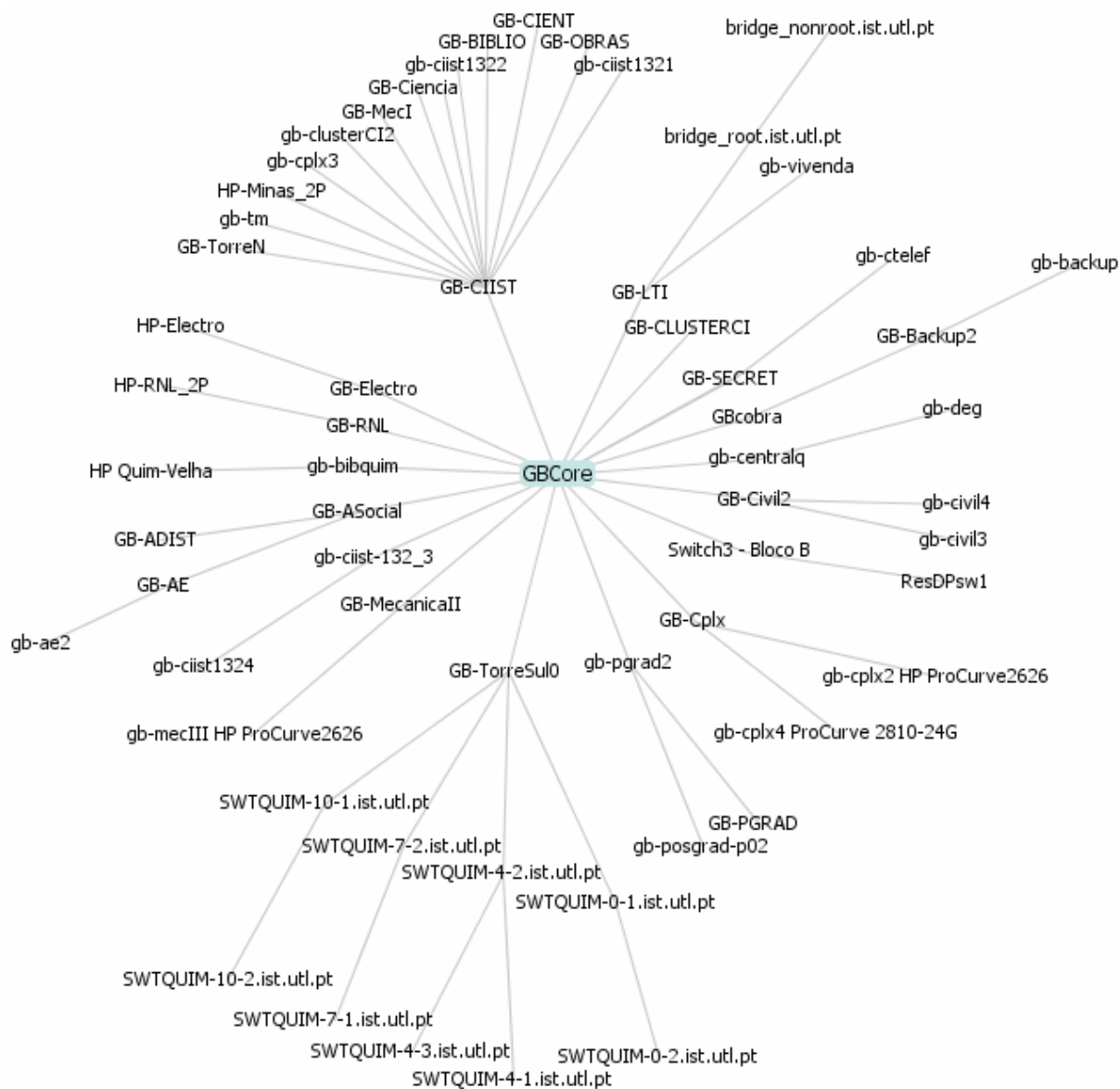


Figura 6.6 – Imagem da nova topologia da rede, em Junho de 2008, representando os equipamentos pelos respectivos nomes configurados.

6.2.2. Localização de Equipamentos na Rede

A partir da informação disponibilizada, é possível deduzir e identificar a localização de um dado equipamento, necessitando apenas de conhecer o seu endereço. Para ilustrar esta funcionalidade, vão apresentar-se os resultados da localização do computador em que foi realizado este projecto, cujo endereço ip é “193.136.132.121” e o respectivo endereço físico “00:04:FF:71:2D”. O equipamento encontra-se ligado à rede do CIIST, mais exactamente no *switch* gigabyte “gb-ciist-132_3”, de endereço ip “192.168.249.219”. Como se pode verificar pela Figura 6.7 e Figura 6.8, a ferramenta desenvolvida identificou correctamente esta informação, localizando o computador tanto a partir do seu endereço ip como do endereço MAC da respectiva placa de rede utilizada.

Localização

Endereço da máquina a localizar:

MAC:

ou IP:

Equipamento ligado a 192.168.249.219 no porto 24

Topologia seguinte:

IP-filho	porto de saída	IP-pai	porto correspondente
192.168.249.219	49	192.168.249.1	6005

Figura 6.7 – Localização de um equipamento na rede, a partir do seu endereço ip.

Localização

Endereço da máquina a localizar:

MAC:

ou IP:

Equipamento ligado a 192.168.249.219 no porto 24

Topologia seguinte:

IP-filho	porto de saída	IP-pai	porto correspondente
192.168.249.219	49	192.168.249.1	6005

Figura 6.8 – Localização de um equipamento na rede, a partir do seu endereço físico.

Esta informação apenas pode ser obtida com exactidão se a topologia for conhecida. Para evidenciar esta situação, vai considerar-se o exemplo da Figura 6.9, que corresponde a parte da topologia obtida previamente.

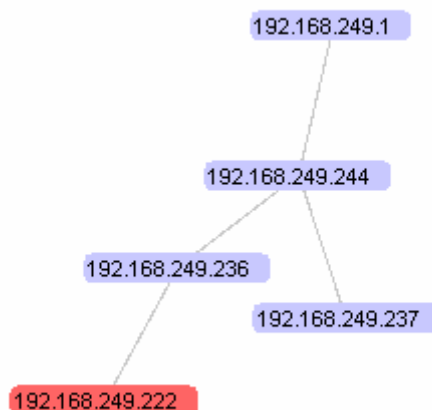


Figura 6.9 – Imagem dinâmica da topologia da rede, centrada no *switch* “192.168.249.222”, contemplando as ligações até ao core da rede (“192.168.249.1”).

Neste caso, escolheram-se para o teste dois equipamentos, com os respectivos endereços ip “193.136.198.20” e “193.136.198.4”. Ambos os equipamentos encontram-se contidos nas AFTs de todos os *switches* da Figura 6.9, não permitindo assim qualquer conclusão sobre a sua localização. No entanto, a partir do conhecimento sobre a hierarquia dos *switches*, das suas ligações e dos respectivos portos utilizados, é possível correlacionar a informação e deduzir a localização dos equipamentos, como se pode observar na Figura 6.10 e na Figura 6.11.

Localização

Endereço da máquina a localizar:

MAC: ou IP:

Equipamento ligado a 192.168.249.236 no porto 24

Topologia seguinte:

IP-filho	porto de saída	IP-pai	porto correspondente
192.168.249.236	49	192.168.249.244	23
192.168.249.244	24	192.168.249.1	1006

Figura 6.10 – Localização de um equipamento na rede, ligado a um *switch* intermédio.

O equipamento “193.136.198.20” encontra-se na AFT relativa ao porto 24 do *switch* “192.168.249.236”. Ora como este porto não corresponde nem ao porto de saída, nem ao porto de

ligação ao *switch* “192.168.249.222”, pode concluir-se que o equipamento “193.136.198.20” apenas pode estar ligado ao *switch* “192.168.249.236”. Com o mesmo algoritmo, pode validar-se a localização de “193.136.198.4”. Este encontra-se na AFT relativa ao porto 1 do *switch* “192.168.249.222”. Sendo este porto diferente do correspondente porto de saída e tendo em conta que este *switch* é o mais afastado do core consoante a hierarquia da rede, é possível deduzir que “193.136.198.4” só pode estar ligado a “192.168.249.222” no porto 1.

Localização

Endereço da máquina a localizar:

MAC: ou IP:

Equipamento ligado a 192.168.249.222 no porto 1

Topologia seguinte:

IP-filho	porto de saída	IP-pai	porto correspondente
192.168.249.222	26	192.168.249.236	7
192.168.249.236	49	192.168.249.244	23
192.168.249.244	24	192.168.249.1	1006

Figura 6.11 – Localização de um equipamento, ligado na periferia da rede.

Pode assim concluir-se que a funcionalidade implementada permite, a partir da informação sobre a topologia da rede, localizar com sucesso qualquer equipamento da rede, desde que este se encontre presente nas AFTs dos *switches*. Os exemplos apresentados confirmam a utilidade desta ferramenta, que possibilita acelerar e facilitar mais uma tarefa de administração de redes.

6.2.3. Detecção de Anomalias

Na intenção de testar a detecção de anomalias, criaram-se diversos modelos de tráfego, correspondendo a diferentes intervalos temporais. Para avaliar a precisão dos modelos identificados, definem-se três tipos de testes, visando validar a detecção de situações irregulares na rede:

- O resultado da detecção automática de anomalias no tráfego real, possibilitada logo após a criação do modelo, pela observação dos alarmes emitidos.
- O resultado da detecção de anomalias criadas manualmente, especialmente concebidas para simular situações anormais na rede. Neste conjunto foram considerados diversos tipos de avarias, resultando num tráfego baixo ou nulo nalguns troços de rede, e situações de tráfego anómalo, tais como picos de carga ou tráfego elevado em períodos de vazio (madrugada ou fim-de-semana), simulando ataques ou aplicações maliciosas.

- O resultado da introdução no modelo de dados aleatórios. Não tendo qualquer tipo de interligação, os dados encontram-se afastados dos padrões normais de funcionamento, resultando, consequentemente, em dados anómalos para o sistema.

Nos testes realizados, é igualmente possível variar o limiar da função de verosimilhança, com o objectivo de estudar qual o valor que melhor se adapta às necessidades, balanceando o número de falsos positivos com o de falsos negativos. Seguidamente, apresentam-se os três modelos criados, juntamente com os resultados obtidos na detecção de anomalias.

Modelo 1:

Neste primeiro modelo, contemplaram-se os dados de duas semanas de utilização da rede do IST, correspondendo a 3718 amostras entre os dias 23 de Junho e 6 de Julho de 2008. A metodologia desenvolvida permitiu a identificação dos parâmetros de uma mistura composta por 5 gaussianas, cujas probabilidades de ocorrência variam entre os 15% e os 28%. Os resultados dos testes realizados podem ser observados na Tabela 6.1e na Tabela 6.2.

Limiar \ Origem dos dados	Tráfego real	Anomalias simuladas	Tráfego aleatório
1.0×10^{-12}	100.0%	28.9%	13.9%
1.0×10^{-8}	99.9%	0.0%	5.0%
1.0×10^{-7}	99.7%	0.0%	3.3%
1.0×10^{-6}	99.2%	0.0%	2.1%
1.0×10^{-3}	77.4%	0.0%	0.0%

Tabela 6.1 – Tráfego considerado normal pelo modelo 1

Limiar \ Origem dos dados	Tráfego real	Anomalias simuladas	Tráfego aleatório
1.0×10^{-12}	0.0%	71.4%	86.1%
1.0×10^{-8}	0.1%	100.0%	95.0%
1.0×10^{-7}	0.3%	100.0%	96.7%
1.0×10^{-6}	0.8%	100.0%	97.9%
1.0×10^{-3}	22.6%	100.0%	100.0%

Tabela 6.2 – Tráfego considerado anómalo pelo modelo 1

Nos resultados apresentados, os falsos negativos correspondem aos valores das colunas “Anomalias simuladas” e “Tráfego aleatório” da Tabela 6.1 (tráfego considerado normal pelo modelo 1). Observa-se que para limiares acima de 1.0×10^{-8} , o erro é inferior a 5%, o que significa que são detectadas mais de 95% das situações anómalas da rede.

Graças a um acompanhamento regular do funcionamento e desempenho da rede do IST, verificou-se que, no desenrolar do trabalho, não ocorreram situações anómalas duradouras e o seu número foi relativamente reduzido, em relação ao funcionamento normal do sistema. Assim sendo, pode considerar-se, como aproximação, que o tráfego amostrado corresponde essencialmente a tráfego normal, sendo a percentagem de anomalias inferior a 0.5%. Baseando-se nesta aproximação, consideram-se falsos positivos as percentagens da coluna “Tráfego real” da Tabela 6.2, quando superiores a 0.5%.

Seguindo o argumentação anterior, pode observar-se que os falsos positivos são desprezáveis, para limiares inferiores a 1.0×10^{-6} (anomalias inferiores a 0.5% no tráfego real). Ora, como explicado anteriormente, é importante manter o número de falsos positivos o mais baixo possível, para que esta ferramenta seja eficaz, mesmo que a detecção se limite às anomalias de maior importância. Pode assim concluir-se que, para este modelo, o limiar ideal é da ordem de 1.0×10^{-7} . Com esta configuração, 97% das anomalias são detectadas, sendo os falsos positivos sensivelmente inexistentes.

Modelo 2:

Este modelo integra os dados das duas semanas que se seguem ao modelo 1, correspondendo a 3949 amostras entre os dias 6 e 20 de Julho de 2008. Foi identificada uma mistura composta por 7 gaussianas, cujas probabilidades de ocorrência variam entre 5% e 22%. Os resultados dos testes realizados podem ser observados na Tabela 6.3 e na Tabela 6.4.

Limiar \ Origem dos dados	Tráfego real	Anomalias simuladas	Tráfego aleatório
1.0×10^{-8}	99.8%	0.0%	2.8%
1.0×10^{-7}	99.6%	0.0%	2.3%
1.0×10^{-6}	98.8%	0.0%	1.1%

Tabela 6.3 – Tráfego considerado normal pelo modelo 2

Mais uma vez se verifica que o limiar 1.0×10^{-7} apresenta os resultados mais satisfatórios. Os falsos positivos são nesse caso desprezáveis, contrariamente aos resultados obtidos com um limiar de 1.0×10^{-6} , enquanto que as anomalias detectadas rondam os 98%, valor superior ao equivalente

calculado com um limiar de 1.0×10^{-8} . Deduz-se assim que, para a rede do IST, o limiar da função de verosimilhança, apresentando os melhores resultados na detecção de anomalias no tráfego, corresponde a 1.0×10^{-7} . Este será então o valor utilizado para os restantes testes, sendo o escolhido como configuração de base para esta ferramenta.

Limiar \ Origem dos dados	Tráfego real	Anomalias simuladas	Tráfego aleatório
1.0×10^{-8}	0.2%	100.0%	97.2%
1.0×10^{-7}	0.4%	100.0%	97.7%
1.0×10^{-6}	1.2%	100.0%	98.9%

Tabela 6.4 – Tráfego considerado anómalo pelo modelo 2

Modelo 3:

Por último, apresenta-se o modelo que tem por base o conjunto de dados disponíveis até à data do teste, correspondendo a 7669 amostras entre os dias 23 de Junho e 20 de Julho de 2008. A mistura é agora composta por 10 gaussianas, cujas probabilidades de ocorrência variam entre 5% e 16%. Os testes foram realizados com o limiar de 1.0×10^{-7} , resultando na Tabela 6.5.

	Tráfego real	Anomalias simuladas	Tráfego aleatório
Tráfego normal	99.7%	0.0%	2.9%
Tráfego anómalo	0.3%	100.0%	97.1%

Tabela 6.5 – Detecção de anomalias a partir do modelo 3

De novo se verifica que mais de 97% das anomalias são identificadas, enquanto os falsos positivos continuam a não ser relevantes. Tal como nos casos anteriores, são detectadas não só as anomalias mais simples, criadas manualmente, mas também a maioria do tráfego aleatório, o que prova a aplicação e abrangência desta funcionalidade.

Neste modelo, verifica-se igualmente que a mistura de gaussianas obtida é mais complexa do que nos casos anteriores, provando assim que esta se adaptou com mais precisão aos dados. O facto deste modelo ter utilizado amostras de cerca de um mês de tráfego pode explicar o sucedido, pois permitiu um melhor reconhecimento de padrões nos dados, nomeadamente dos que estão relacionados com as diferenças observáveis entre os dias da semana. Pelo contrário, os modelos 1 e 2, mais simples, basearam-se em apenas duas semanas de tráfego, não evidenciando assim todos os padrões existentes nos dados, resultando em misturas menos específicas.

Os resultados apresentados permitem concluir que o sistema de aprendizagem não supervisionado desenvolvido possibilita a criação de um modelo de tráfego que caracterize o funcionamento normal do sistema. A partir deste mecanismo, possibilita-se a detecção de situações de funcionamento anómalo e assim lançar alertas aos administradores de sistemas, minimizando os efeitos negativos das irregularidades observadas.

6.2.4. Identificação de Anomalias

Para terminar a apresentação de resultados, vão identificar-se algumas das anomalias reais detectadas pela ferramenta, utilizando para o efeito os parâmetros do modelo 3. Como se pode observar na Figura 6.12, foram detectadas 13 anomalias durante o mês testado, algumas das quais se prolongaram por mais de uma amostra.

Anomalias detectadas

Data de início (formato: "dd/mm/aa hora:minuto"):

Data de final (formato: "dd/mm/aa hora:minuto"):

Datas em que foram detectadas anomalias no tráfego:

Foram detectadas anomalias de tráfego nas seguintes datas:

- > 24/06/08 11:37
- > 25/06/08 16:32
- > 27/06/08 15:27
- > 30/06/08 13:22
- > 30/06/08 13:27
- > 30/06/08 17:02
- > 07/07/08 16:48
- > 07/07/08 17:18
- > 07/07/08 17:23
- > 07/07/08 18:03
- > 07/07/08 18:08
- > 07/07/08 18:13
- > 07/07/08 18:18
- > 10/07/08 12:44
- > 14/07/08 15:51
- > 15/07/08 10:41
- > 17/07/08 09:46
- > 17/07/08 17:06
- > 18/07/08 19:01
- > 18/07/08 19:06
- > 18/07/08 19:21
- > 18/07/08 19:26
- > 19/07/08 04:31

Figura 6.12 – Detecção de anomalias baseada no modelo 3.

Depois de detectada uma anomalia, é possível utilizar as funcionalidades da ferramenta para tentar identificar alguma informação sobre a situação irregular. Na Figura 6.13, apresentam-se os parâmetros escolhidos para a elaboração do modelo. A partir desta informação, pode confinar-se a análise dos valores de tráfego aos troços de rede amostrados, na intenção de pesquisar sobre as causas da situação anómala descoberta.

INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

Introdução
Descoberta
Polling
Modelo
Anomalias
Informação
VLANs
Topologia
Autor

Anomalias na rede

Informação sobre o modelo

Foram utilizadas 7 dimensões neste modelo.

O modelo foi criado a partir das seguintes entradas de dados:

- > Dia da semana;
- > Hora da amostragem;
- > Tráfego nos seguintes portos do core: [1001, 2008, 2009, 2010, 5010].

O limiar configurado para a detecção de anomalias foi o seguinte: 1.000000e-007.

Anomalias detectadas

Data de início (formato: "dd/mm/aa hora:minuto"):

Data de final (formato: "dd/mm/aa hora:minuto"):

Datas em que foram detectadas anomalias no tráfego:

©2008, Instituto Superior Técnico. Todos os direitos reservados.

Figura 6.13 – Informação sobre os parâmetros do modelo de tráfego.

Nem todas as anomalias são facilmente identificáveis, pois podem ser o resultado de uma combinação de factores e não de uma irregularidade num só troço de rede. No entanto, nalguns casos mais simples, é possível observar claramente a perturbação no tráfego, responsável pela situação detectada.

Na Figura 6.14, encontra-se evidenciada a anomalia de dia 15 de Julho, detectada cerca das 10h41, como se pode validar pela Figura 6.12. O gráfico apresentado indica um volume de tráfego de saída anormalmente elevado no porto 1001 do *router* central, observado na manhã do dia 15. Verifica-se adicionalmente que esta situação é singular, não se repetindo em qualquer um dos restantes

dados amostrados. Podemos assim concluir que a detecção de anomalias identificou correctamente a irregularidade e que esta teve origem no troço de rede ligado ao porto 1001 do core, localizando assim a origem da ocorrência.

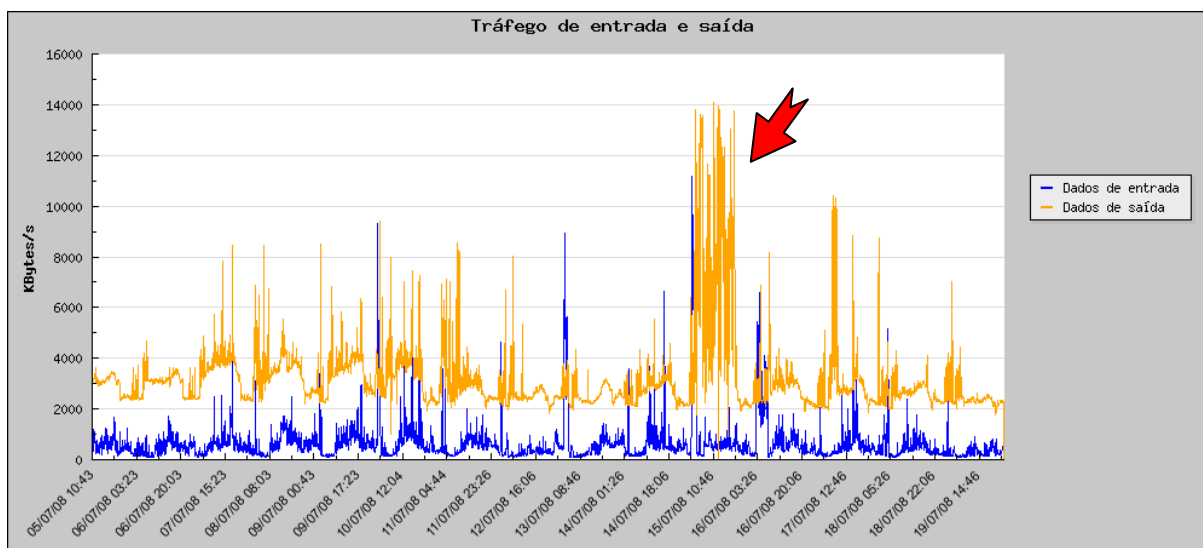


Figura 6.14 – Tráfego no porto 1001 do *router* central, evidenciando a anomalia de dia 15 de Julho.

Num outro exemplo, destaca-se a anomalia de dia 19 de Julho, detectada cerca das 04h31. Neste caso, observa-se um tráfego de entrada sensivelmente nulo, situação inédita no porto 2008 do *router* central, tal como se pode observar pelo andamento do gráfico da Figura 6.15. Estando a anomalia confinada à sub-rede indicada, seria importante procurar o estado dos equipamentos ligados ao porto 2008 do core, na intenção de despistar qualquer avaria existente. No caso de não serem identificados problemas operacionais, pode deduzir-se que a redução no tráfego foi devida a um período de vazio, obtido pela conjugação da hora (madrugada) com o início das férias (finais de Julho).

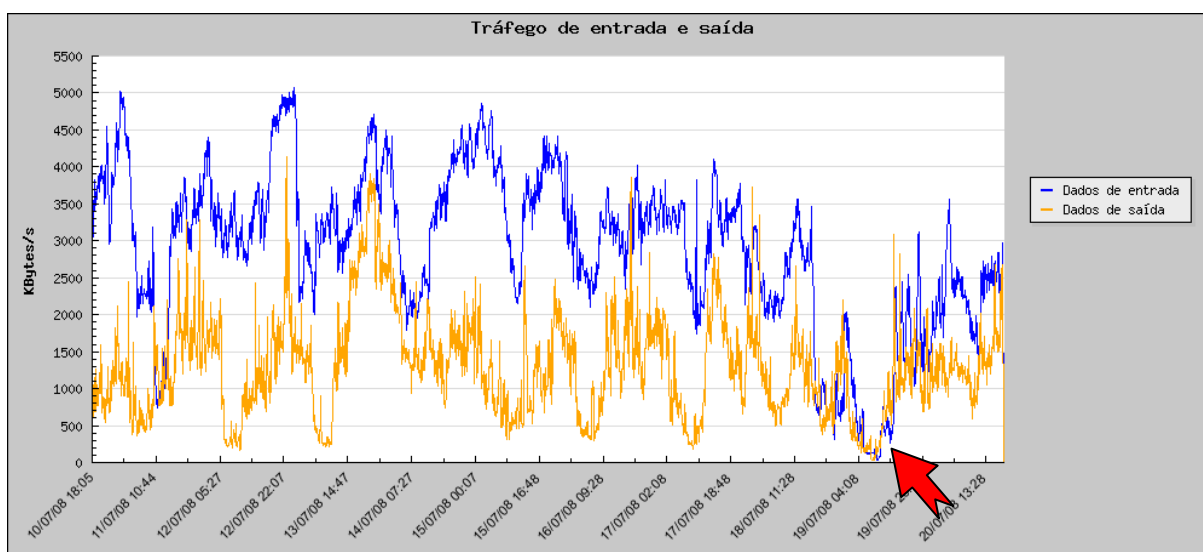


Figura 6.15 – Tráfego no porto 2008 do *router* central, destacando a anomalia de dia 19 de Julho.

Pode assim concluir-se que a ferramenta desenvolvida permite, não só uma detecção de anomalias no funcionamento da rede, mas também a localização da proveniência da anomalia, acelerando, dessa forma, o processo de resolução da mesma. Embora os exemplos apresentados sejam relativos a eventos passados, esta funcionalidade pode ser utilizada em tempo real, com o objectivo de providenciar uma resposta rápida e útil para a gestão da rede do IST.

7. Conclusões

7.1. Observações Finais

Este projecto teve como principal objectivo a criação de uma ferramenta que facilitasse e acelerasse as tarefas de administração e gestão de redes, disponibilizando uma nova gama de aplicações e proporcionando um importante acréscimo na segurança da informação. Teve como ponto de partida o estudo das tecnologias e plataformas existentes actualmente, com o intuito de decidir sobre a orientação e caminho a seguir. Escolheu-se criar uma nova aplicação, de raiz, adaptada às características de uma rede em estrela, centralizada, como a do IST. Deste modo, foi possível ultrapassar os obstáculos encontrados em programas comerciais, dispendiosos e muito proprietários, direccionando o trabalho no sentido da flexibilidade e expansão.

Seguidamente, foi desenvolvido um novo algoritmo que, mediante a ajuda de diversas heurísticas, pudesse realizar uma descoberta automática da topologia da rede. O algoritmo baseia-se na recolha, por SNMP, da informação existente nos equipamentos de rede. A análise e o tratamento dos dados obtidos permitem a dedução das ligações entre equipamentos e assim concluir sobre a topologia global existente.

Os resultados obtidos acompanharam correctamente as modificações na rede do IST, confirmando assim o funcionamento da ferramenta desenvolvida, indo de encontro aos objectivos traçados. Graças a essa informação, tornou-se possível implementar operações de gestão centralizadas, tais como a localização imediata de equipamentos na rede, minimizando assim as tarefas manuais pouco eficientes.

Paralelamente, foi elaborada uma base de dados personalizada e exportável, sendo preenchida com informação sobre o estado da rede, nomeadamente o tráfego existente e o tempo de resposta dos equipamentos. O seu preenchimento efectua-se graças a uma interrogação periódica à rede, de forma a manter os dados constantemente actualizados. Junta-se igualmente a esta base de dados informação sobre a configuração da rede, tal como a descrição dos equipamentos e as VLANs que lhes estão associadas. A partir desta monitorização, obtém-se um importante repositório de informação, cuja análise e exploração abrem uma vasta sucessão de possibilidades.

A partir da informação registada, foi possível dotar a ferramenta de um mecanismo de aprendizagem não supervisionada, resultando numa modelação do tráfego usual na rede. A comparação do tráfego real com os valores do modelo permite uma avaliação sobre o estado global do sistema, reflectindo potenciais não conformidades, tais como falhas nos equipamentos ou ataques à rede. Complementa-se assim a ferramenta desenvolvida com uma funcionalidade de detecção de anomalias no tráfego,

mesmo quando desconhecidas, o que facilita significativamente a tarefa de vigilância e controlo da rede do IST.

Os resultados foram mais uma vez de encontro às expectativas, possibilitando a identificação de anomalias variadas, não deixando de minimizar o número de falsos positivos. Foi possível distinguir as oscilações normais de tráfego daquelas provocadas por uma falha ou uma sobrecarga na rede, constituindo assim um bom indicador sobre o estado de funcionamento dos sistemas.

As funcionalidades implementadas contribuem assim, não só para um melhoramento e uma aceleração de diversas tarefas de gestão, mas também para um aumento da segurança na rede, reduzindo o tempo de detecção e reacção a situações irregulares. Conclui-se inserindo o trabalho realizado num processo de evolução global, orientado no sentido da automatização, das operações de controlo e administração de redes, e do aumento de segurança dos sistemas.

7.2. Aplicações Futuras

Destaca-se como primeira expansão ao projecto, o seu alargamento a outros tipos de equipamentos, tais como *Access Points* (AP) ou *routers* e *switches* de outras redes. Esta informação permitiria generalizar a descoberta da topologia e melhorar a monitorização, acedendo, assim, a mais detalhes sobre as redes em questão. Quanto mais informação estiver na posse dos gestores, maior é o controlo da rede e, consequentemente, mais eficaz é a sua administração.

No caso desta ferramenta ser integrada na gestão corrente do IST, podem ser automatizadas mais algumas operações, na intenção de tornar a ferramenta mais intuitiva e eficiente. A interacção com a ferramenta pode, por exemplo, ser integrada numa consola de gestão global, juntamente com as funcionalidades de outras ferramentas. Pode ser igualmente melhorada a geração de sinalização, enviando alarmes em tempo real aos administradores de rede, por correio electrónico ou mensagem escrita, aquando da ocorrência de alguma irregularidade.

Existe ainda a possibilidade de interpretar essa sinalização, de forma automática e direccionada, por uma ferramenta que faculte operações de intervenção remota na rede. Numa situação em que seja localizada uma anomalia grave no funcionamento da rede, necessitando de uma intervenção rápida, é possível bloquear temporariamente a sub-rede responsável, encerrando o porto do *switch* ao qual esta se encontra ligada. Este processo pode ser realizado automaticamente graças, por exemplo, ao protocolo SNMP que permite, em modo de “escrita”, alterar as configurações nos equipamentos. O objectivo consiste em dotar as redes de funcionalidades que permitam uma gestão autónoma, num crescendo de eficácia, minorando assim a necessidade de intervenção humana.

Por último, pode ser aprofundado o estudo sobre a identificação de modelos de tráfego, procurando outras abordagens ou metodologias. Algoritmos mais complexos podem ser adaptados a esta ferramenta, no intuito de se identificarem quais os modelos que melhor se adequam aos dados reais. Como sugestão, deixa-se a possibilidade de analisar outros métodos de comparação entre modelos, utilizando por exemplo o *Minimum Description Length* (MDL), ou outros algoritmos para a procura dos parâmetros dos modelos. Existem, nomeadamente, métodos estatísticos como o Método de Monte Carlo baseado em Cadeias de Markov (MCMC) [DaYi03], cuja abordagem se opõe aos tradicionais algoritmos determinísticos, tais como o EM, proporcionando assim outro tipo de resultados.

A. Manual técnico

A ferramenta desenvolvida encontra-se correctamente configurada e a funcionar no rede do IST. Como já referido, todas os parâmetros variáveis da configuração estão definidos num ficheiro próprio, para fácil consulta e alteração, cujo conteúdo pode ser observado na Figura A.1.

Do mesmo modo, a interface Web usa um ficheiro de configuração responsável, nomeadamente, pelo formato e tipo de comunicação usado na interacção com a ferramenta. Todos estes parâmetros encontram-se na Figura A.2.

O restante das configurações, assim como todo o código desenvolvido, encontram-se na versão electrónica do projecto, na intenção de não sobrecarregar este relatório.

Este projecto foi realizado no Windows XP, devido às experiências iniciais realizadas no HP-Open-View, cuja única versão disponível era executada nesse sistema operativo. No entanto, toda a ferramenta desenvolvida pode ser utilizada noutros ambientes (como por exemplo o Linux), dado terem sido escolhidas linguagem suportadas pela maioria dos sistemas existentes (Python, Java, PHP). Os pacotes e bibliotecas utilizados (Nemesis, MySQL, MySQLdb, Net-SNMP, Jpgraph, Pymml) foram também escolhidos por serem multi-plataformas, assim como o servidor Apache, cuja configuração pode ser exportada para qualquer outra instalação. Desta forma, garante-se a flexibilidade e a possibilidade de expansão, princípios essenciais no desenrolar deste trabalho.

```
# Caminho para o ficheiro contendo a lista dos ips para a descoberta inicial
fich_ip = r'C:\py\ips.txt'

# Community string por omissão (no caso de esta não se encontrar no ficheiro que se segue)
comm_str_default = 'public'

# Caminho para o ficheiro contendo a lista dos ips cuja community string não é a de omissão
fich_comm = r'C:\py\community_strings.txt'

# Intervalo no polling ao equipamento
t_polling = 5 # minutos

# Ficheiro de emulação de pipe para comandar o polling por script externo (ou a interface Web)
comando = 'fim_polling.txt'

# Tempo entre a criação das threads (atraso na descoberta inicial mas menos sobrecarga na rede)
t_atraso_thread = 2 # segundos

# Caminho para o ficheiro em que se encontra a lista de instruções sql para limpar as tabelas da
# base de dados (ou seja contendo os nome das tabelas da base de dados)
fich_sql = r'C:\BD\limpar_tabelas.sql'

# Caminho para o programa de spoofing de pacotes icmp
spoofing_prg = r"C:\tools\nemesis"
```

```

# Spoofing na rede
efectuar_spoofing = 1 # por a 0 para desligar e a 1 para ligar

# Intervalo de tempo para novo spoofing à rede: para garantir que as tabelas AFT estão sempre
# o mais completo possível
# No geral, tabelas AFT são limpas e a cada 300s (5min)
t_spoofing = 4 # minutos

# Endereço IP inexistente na rede para fazer spoofing
ip_inexistente = '192.168.249.20'

# Timeout do snmpwalk
timeout_snmpwalk = 3 # (s)

# Ligação à base de dados
host_ = "localhost"
user_ = "gestor"
passwd_ = "g4st0r"
db_ = "rede_alameda"

# Fazer as queries na máquina de suporte em vez de utilizar o pc do tfc
suporte = 1 # 1 ou 0 consoante se use ou não o suporte respectivamente
cliente_ssh = "ssh"
servidor_suporte = "mlpc@suporte.ist.utl.pt"

# Número a colocar na base de dados quando o porto é desconhecido
porto_desconhecido = -1

# Número a colocar na base de dados no campo porto, quando se trata do chassis
# para distinguir dos portos físicos
chassis = -2

# Router de saída
router_saida = "192.168.249.1"

# Localização e nomes dos ficheiros de entrada do Prefuse
prefuse_path = "C:\\xampp\\htdocs\\applets\\prefuse\\"
topologia_nomes = "topologia_ist_nomes.xml"
topologia_ips = "topologia_ist_ips.xml"

# Formato da data utilizada
formato_data = "%d/%m/%y %H:%M"

# Dados para o scan de switches online (scan_sw)
comms = ['public','publicsw','read_mlpc']
base_ip = "192.168.249."
procura_sw_online = 0 # 1 ou 0 consoante se faça ou não uma nova procura de switches online
na rede

# Dados para a criação do modelo e detecção de anomalias na rede
ficheiro_modelo = r'C:\py\modelo.bin'
ficheiro_dados_modelo = r'C:\py\dados_modelo.bin'
dim = 7 # dimensao do modelo (ex: 6 = 4 portos do core + hora + dia da semana)
portos_escolhidos = [1001, 2008, 2009, 2010, 5010]
#portos escolhidos - têm de ser colocados por ordem crescente
dados_escolhidos = [ 1, 0, 0, 0, 1]
# relativo aos portos_escolhidos, a 0 se bytes de entrada ou a 1 se bytes de saída
limiar = 1.0e-7 # Limiar abaixo do qual os dados são considerados anómalos

```

Figura A.1 – Ficheiro de configuração da ferramenta criada.

```

// Caminho onde se encontram os scripts python
$path_script_queries = 'C:\py\queries.py';
$path_script_polling = 'C:\py\polling.py';
$path_script_exploracao_inicial = 'C:\py\exploracao_inicial.py';
$path_script_modelo_trafego = 'C:\py\modelo.py';
$path_script_queries_modelo = 'C:\py\anomalias_modelo.py';

// Caminho onde se encontra o programa em C para fazer fork, exec
// e em seguida retornar o processo pai
$path_executa = 'C:\tools\executa python';

// Argumentos a passar aos scripts python para executar os diferentes tipos de queries
$get_sw = 'get_sw';
$get_rtt_sw = 'get_rtt_sw';
$get_trafego_sw = 'get_trafego_sw';
$get_localizacao = 'get_localizacao';
$get_topologia_sw = 'get_topologia_sw';
$get_vlans = 'get_vlans';
$get_vlans_ip = 'get_vlans_ip';
$get_ips_vlan = 'get_ips_vlan';
$get_info_modelo = 'get_info_modelo';
$get_anomalias = 'get_anomalias';

// Ligação à base de dados (para poder conhecer o estado da descoberta e do polling)
$user="leitura";
$password="le1tur@";
$database="rede_alameda";
$address = '127.0.0.1';

// Ficheiro de emulação de pipe para comandar o polling, efectuado por um script externo
$File_comando_polling = "../polling/fim_polling.txt";

// Nome do ficheiro de log contendo o decorrer da descoberta da rede
$log_descoberta = 'log_descoberta.txt';

// Caminho da Applet Prefuse
$include_prefuse = "prefuse.demos.applets.";
$path_prefuse = "../applets/prefuse/";

// Classes Java para visualização da topologia
$prefuse_topologia_radial_ips = "Topologia_radial_ips.class";
$prefuse_topologia_radial_nomes = "Topologia_radial_nomes.class";
$prefuse_topologia_ips = "Topologia_ips.class";
$prefuse_topologia_nomes = "Topologia_nomes.class";

```

Figura A.2 – Ficheiro de configuração da interface Web.

B. Manual utilizador

Escolheu-se como interface com o utilizador uma página Web intuitiva, pela sua simplicidade e facilidade de acesso. Apresenta-se em seguida um pequeno manual tendo, por principal finalidade, a caracterização das funcionalidades disponibilizadas.

Como se pode verificar na Figura B.1, as diferentes operações são acedidas por um menu à esquerda, cujas possibilidades vão ser exploradas seguidamente.

Descoberta Inicial

Esta secção (Figura B.1) indica a data da última descoberta da rede efectuada e possibilita iniciar, novamente, a operação. No caso de ser iniciada uma nova descoberta, torna-se possível observar o estado da mesma, em que os diferentes passos são apresentados aquando da sua execução.

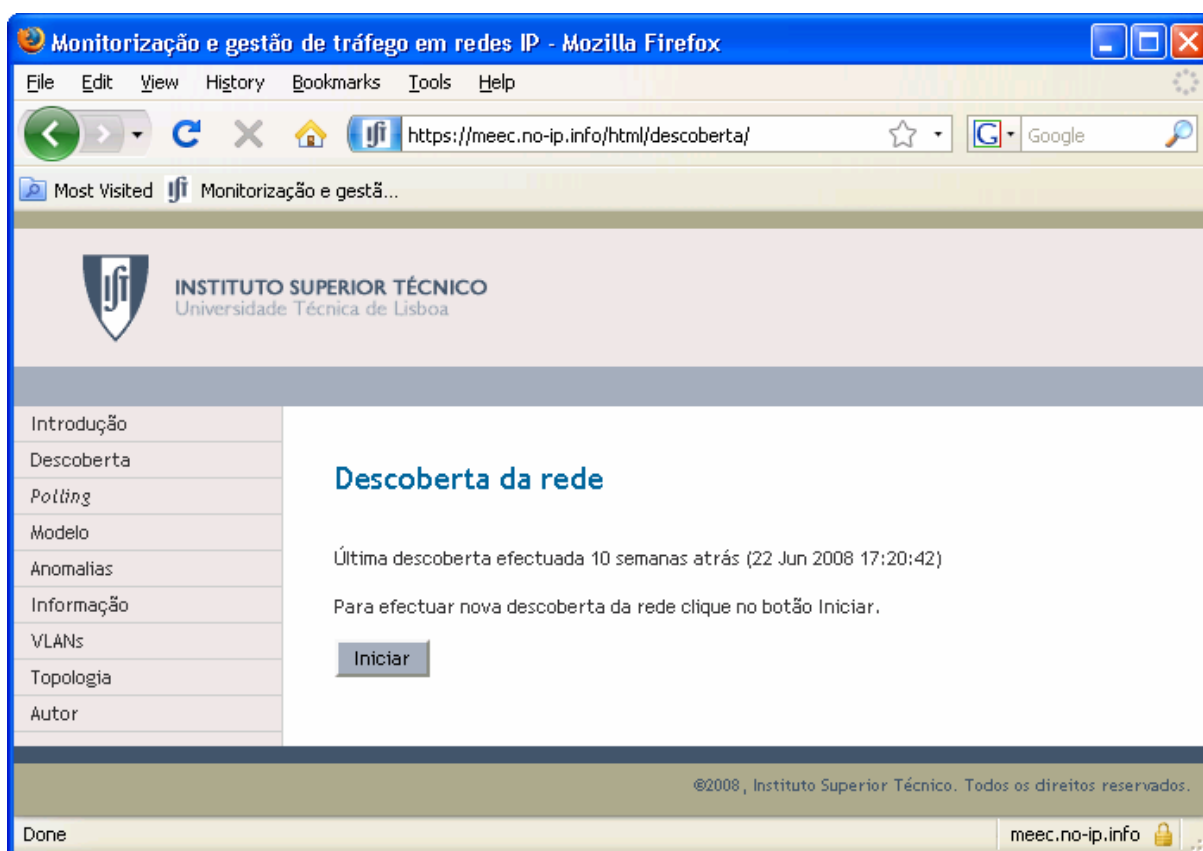


Figura B.1 – Descoberta da rede, na interface Web.

Polling

Esta página apenas indica e permite alterar o estado do *polling*. Se, com o *polling* activo, for enviado o comando “parar” a meio de uma recolha de informação na rede, esta será finalizada e só depois o *polling* será considerado desligado. Esta situação ocorre para não preencher a base de dados com dados parciais, podendo originar más interpretações ou falhas na integridade dos resultados.

Modelo de Tráfego

Tal como no caso do *polling*, esta página tem por funcionalidade permitir a criação de um novo modelo de tráfego. Após esta operação estar concluída, as anomalias podem ser verificadas na página “Anomalias”, tal como se pode observar na Figura B.2.

Anomalias detectadas

Data de início (formato: "dd/mm/aa hora:minuto");

23/06/08 18:00

Data de final (formato: "dd/mm/aa hora:minuto");

20/07/08 18:00

Datas em que foram detectadas anomalias no tráfego:

Pesquisar

Foram detectadas anomalias de tráfego nas seguintes datas:

-> 24/06/08 11:37

-> 25/06/08 16:32

-> 27/06/08 15:27

-> 30/06/08 13:22

-> 30/06/08 13:27

-> 30/06/08 17:02

-> 07/07/08 16:48

-> 07/07/08 17:18

-> 07/07/08 17:23

-> 07/07/08 18:03

-> 07/07/08 18:08

-> 07/07/08 18:13

-> 07/07/08 18:18

-> 10/07/08 12:44

-> 14/07/08 15:51

-> 15/07/08 10:41

-> 17/07/08 09:46

-> 17/07/08 17:06

-> 18/07/08 19:01

-> 18/07/08 19:06

-> 18/07/08 19:21

-> 18/07/08 19:26

-> 19/07/08 04:31

Figura B.2 – Anomalias detectadas na rede.

Informação

Nesta secção (Figura B.3) é possível não só obter diversos tipos de informação sobre a rede, tal como listar os equipamentos geridos juntamente com as suas descrições, mas também efectuar algumas operações de gestão.

Informação da rede

Informação geral

Lista de *switches* monitorizados da rede:

Tráfego

Informação sobre o tráfego e o estado dos *switches*.

IP: Porto:

Data de início (formato: "dd/mm/aa hora:minuto"):

Data de final (formato: "dd/mm/aa hora:minuto"):

Tempo de resposta (RTT): Tráfego de entrada:

Localização

Endereço da máquina a localizar:

MAC: ou IP:

Figura B.3 – Informação sobre a rede, na interface Web.

Introduzindo o endereço IP de um *switch*, faculta-se o histórico do seu tempo de resposta ou do seu tráfego associado. Sendo esta informação muito extensa, no geral, é possível limitar a procura a um intervalo temporal e, no caso do tráfego, a um porto específico. Deste modo, obtém-se a informação desejada, de uma forma clara e objectiva. No início da listagem do tráfego, existe igualmente um *link* para a visualização gráfico do tráfego seleccionado, resultando nas imagens já apresentadas ao longo deste trabalho (ver capítulo 6.2.4).

Outra funcionalidade consiste na localização de um equipamento, tanto pelo seu endereço MAC como pelo de rede (IP). No caso das ligações encontradas serem com um *switch* cujo porto utilizado não é conhecido, será apresentada a notação N/A (*Non Available*).

Em qualquer altura, para efectuar uma nova operação, basta pressionar o botão “Nova procura” e todos os campos e listagens serão imediatamente limpos.

VLANs

A informação anterior é complementada pelo conhecimento providenciado pelas VLANs. Adicionalmente a apresentação da lista de VLANs existentes na rede, possibilita-se uma procura por equipamento ou por VLAN. Na primeira situação, listam-se todas as VLANs identificadas no *switch* indicado, juntamente com os portos em que estão associadas, ou, no caso do campo “porto” ser introduzido, apenas as VLANs relacionadas à interface correspondente. Na segunda procura, retorna-se o alcance das VLANs, indicando quais os portos e *switches* que as utilizam.

Informação sobre as VLANs

Procura de VLANs

Lista de VLANs existentes na rede:

Procura por endereço IP

IP: Porto:

VLANs associadas ao *switch* indicado:

Procura por VLAN

VLAN ID: ou o seu nome:

Switches contendo a VLAN indicada:

Figura B.4 – Informação sobre as VLANs, na interface Web.

Topologia

Por último, esta secção apresenta a visualização da imagem da rede, cujo resultado já foi exemplificado no capítulo 6.1.2. Podem listar-se as ligações entre *switches* sob a forma textual ou escolher o formato das suas representações e observar o resultado gráfico fornecido pela Applet Prefuse (Figura B.5).

Topologia da rede

Ligações entre *switches*

Informação sobre a topologia de rede (ligações *switches*):

Imagem da rede

Representação dos *switches*: ☒ nome ☐ ip

Apresentação das ligações: ☒ dinamica ☐ radial

Imagem da topologia de rede:

Figura B.5 – Topologia de rede, na interface Web.

Referências

[BBGR03] Y. Bejerano, Y. Breitbart, M. Garofalakis, and R. Rastogi, "Physical topology discovery for large multi-subnet networks", in *Proc. IEEE INFOCOM*, 2003, pp. 342–352.

[BDFo04] Richard Black, Austin Donnelly, Cedric Fournet, "Ethernet Topology Discovery without Network Assistance", *Proceedings of 12th IEEE International Conference on Network Protocols (ICNP'04)*, 2004, pp. 328-339.

[BGJM04] Yuri Breitbart, Minos Garofalakis, Ben Jai, Cliff Martin, Rajeev Rastogi, and Avi Silberschatz, "Topology Discovery in Heterogeneous IP Networks: The NetInventory System", *IEEE/ACM Transactions on Networking*, Vol. 12, No. 3, June 2004, pp. 401-414.

[BGMR00] Y. Breitbart, M. Garofalakis, C. Martin, R. Rastogi, S. Seshadri, and A. Silberschatz, "Topology discovery in heterogeneous IP networks," in *Proc. IEEE INFOCOM*, 2000, pp. 265–274.

[BiJo00] A. Bierman and K. Jones, "Physical Topology MIB", Internet RFC-2922, Sept. 2000 (<http://www.ietf.org/rfc/rfc2922.txt>).

[Bish95] Christopher M. Bishop, *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, UK, 1995.

[CFSD90] J. Case, M. Fedor, M. Schoffstall, and J. Davin, "A simple network management protocol (SNMP)," RFC-1157, May 1990 (<http://www.ietf.org/rfc/rfc1157.txt>).

[CoMu04] G. Cormode, S. Muthukrishnan, "What's new: Finding significant differences in network data streams," in *Proc. of IEEE Infocom*, 2004.

[DaYi03] Ian Davidson, Ke Yin, "Message Length Estimators, Probabilistic Sampling and Optimal Prediction", *DIMACS Workshop on Complexity and Inference*, 2003.

[IEEE04] IEEE Computer Society, *IEEE Std 802.1D-2004, IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Bridges*, IEEE Standard, 2004.

[Gals06] Ethan Galstad, "Nagios", *Open source host, service and network monitoring program*, versão 2.4, Maio 2006 (<http://www.nagios.org/>).

[GHYC06] Jun Gao, Guangmin Hu, Xingmiao Yao, Rocky K. C. Chang, "Anomaly Detection of Network Traffic Based on Wavelet Packet", *IEEE*, 2006.

[GrNa04] Mark Grimes , Jeff Nathan , “Nemesis”, *Command-line network packet crafting and injection tool*, versão 1.4, Outubro 2004 (<http://nemesis.sourceforge.net/>).

[Harr05] Paul Harrison, “PyMML”, Biblioteca de estimadores MML para o Python, versão 0.5, 2005 (<http://www.logarithmic.net/pfh/pymml>).

[Heer06] Jeffrey Heer, “Prefuse”, *A Java-based toolkit for building interactive information visualization applications*, Maio 2006, (<http://sourceforge.net/projects/prefuse/>).

[HPOV] HP, “HP OpenView Network Node Manager”, (<http://www.novadigm.com/>).

[HPOV03a] Hewlett-Packard Development Company, *Managing Your Network with HP OpenView Network Node Manager*, HP, 2003.

[HPOV03b] Hewlett-Packard Development Company, *Reporting and Data Analysis with Network Node Manager*, HP, 2003.

[JiPa03] Jun Jiang, Symeon Papavassiliou, “A Network Fault Diagnostic Approach Based on a Statistical Traffic Normality Prediction Algorithm”, *GLOBECOM*, IEEE, 2003, pp. 2918-2922.

[Jpgr07] “JpGraph”, biblioteca PHP para criação de gráficos, versão 1.22, Outubro 2007 (<http://www.aditus.nu/jpgraph/index.php>)

[KSZC03] B. Krishnamurthy, S. Sen, Y. Zhang, Y Chen, “Sketch-based Change Detection: Methods, Evaluation, and Applications”, *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, 2003.

[KuRo04] James F. Kurose, Keith W. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet*, Addison Wesley, USA, 2004.

[LLLi07] Jun Lv, Xing Li, Tong Li, “The New Detection Algorithms for Network Traffic Anomalies”, *Proceedings of the Sixth International Conference on Networking (ICN'07)*, IEEE, 2007.

[LOGr01] B. Lowekamp, D. R. O'Hallaron, and T. R. Gross, “Topology Discovery for Large Ethernet Networks”, in *Proceedings of ACM SIGCOMM*, San Diego, California, Aug. 2001.

[LPZL05] Yuzhao Li, Changxing Pei, Changhua Zhu, Jiandong Li, “An Algorithm for Discovering Physical Topology in Single Subnet IP Networks”, in *Proceedings of 19th International Conference on Advanced Information Networking and Applications (AINA'05)*, Volume 2, 2005, pp. 351-354.

[LuAs03] Mark Lutz, David Ascher, *Learning Python*, O'Reilly, USA, 2003.

[LWWC01] Hwa-Chun Lin, Yi-Fan Wang, Chien-Hsing Wang, Chien-Lin Chen, "Web-based Distributed Topology Discovery of IP Networks", *Proceedings of 15th International Conference on Information Networking (ICOIN'01)*, 2001, pp. 857-862.

[MaLo] Ricardo David Martins, Rui Diogo Lopes, *Estudo de Uma Ferramenta de Gestão de Redes Nagios*, ISCTE, Portugal.

[MaSc05] Douglas R. Mauro, Kevin J. Schmidt, *Essential SNMP*, O'Reilly, USA, 2005.

[Maxi90] Roy A. Maxion, "Anomaly Detection for Diagnosis", *IEEE*, 1990.

[McRo91] K. McCloghrie, M. Rose, "Management Information Base for Network Management of TCP/IP-based internets: MIB-II", RFC 1213, Março 1991 (<http://tools.ietf.org/html/1213>).

[Mysq06] "MySQLdb", Biblioteca para interacção do Python com o Mysql, versão 1.2.1, Abril 2006 (<http://sourceforge.net/projects/mysql-python>).

[Nets06] "Net-SNMP", Pacote *Open Source* contendo diversos comandos para efectuar operações SNMP tanto para Linux como para Windows, Janeiro 2006 (<http://net-snmp.sourceforge.net/>).

[Post81] J. Postel, "Internet Control Message Protocol", RFC 792, Setembro 1981 (<http://www.ietf.org/rfc/rfc0792.txt>).

[Pysn06] "Pysnmp", Motor SNMP para Python, versão 2.0.9, Maio 2006 (<http://pysnmp.sourceforge.net/>).

[RoMc90a] M. Rose, K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based internets", RFC 1155, Março 1990 (<http://tools.ietf.org/html/1065>).

[RoMc90b] M. Rose, K. McCloghrie, "Management Information Base for Network Management of TCP/IP-based internets", RFC 1156, Março 1990 (<http://tools.ietf.org/html/1066>).

[ShWe49] Claude Elwood Shannon, Warren Weaver, *The mathematical theory of communication*, Urbana: University of Illinois Press, USA, 1949.

[SoKa91] T. Socolofsky, C. Kale, "A TCP/IP Tutorial", RFC 1180, Janeiro 1991 (<http://www.ietf.org/rfc/rfc1180.txt>).

[Stot02a] D. T. Stott, "Layer-2 path discovery using spanning tree mibs," Avaya Labs Research, Avaya Inc., Basking Ridge, NJ, Tech. Rep. ALR-2002-004, 2002.

[Stot02b] D. T. Stott, "Snmp-based layer-3 path discovery," Tech. Rep. ALR-2002-005, Avaya Labs Research, Avaya Inc., Basking Ridge, NJ, 2002.

[SWSh05] Yantao Sun, Zhimei Wu, Zhiqiang Shi, "The physical topology discovery for Switched Ethernet based on Connections Reasoning Technique", in *Proceedings of International Symposium on Communications and Information Technologies 2005*, Vols. 1 and 2, 2005, pp. 42-45.

[Tane02] Andrew S. Tanenbaum, *Computer Networks*, Prentice Hall, USA, 2002.

[Vaza04] Teresa Maria Sá Ferreira Vazão Vasques, *Texto de Apoio à Gestão de Redes e de Sistemas Distribuídos*, Instituto Superior Técnico, Julho 2004.

[Wall05] C.S. Wallace, *Statistical and Inductive Inference by Minimum Message Length*, Springer, USA, 2005.